DOCUMENT RESUME

ED 089 663 IR 000 373

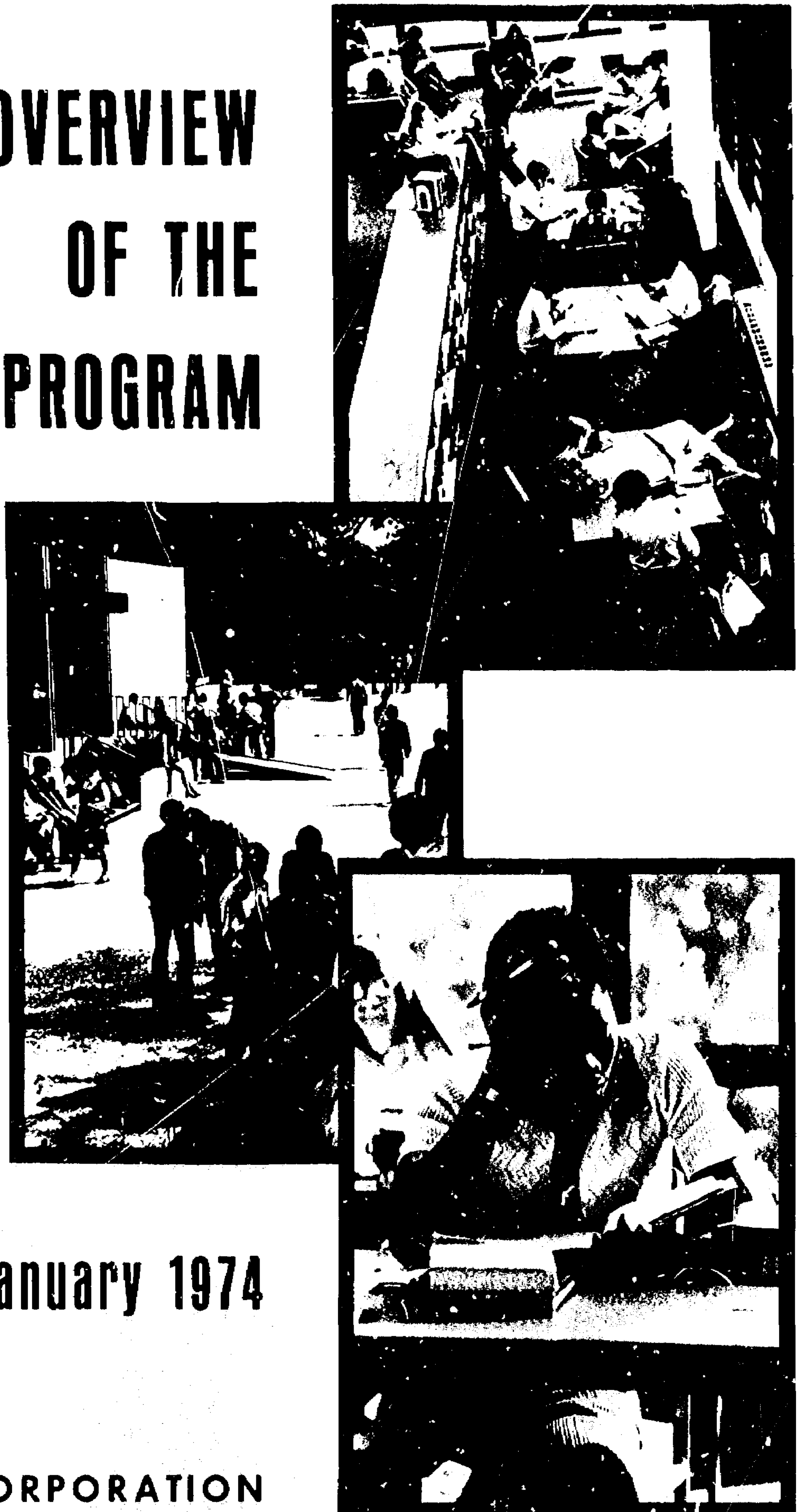
TITLE	An Overview of the TICCIT Program.
INSTITUTION	Mitre Corp., McLean, Va.
SPONS AGENCY	National Science Foundation, Washington, D.C.
REPORT NO	M74-1
PUB DATE	Jan 74
NOTE	71p.
EDRS PRICE	MF-$0.75 HC-$3.15 PLUS POSTAGE
DESCRIPTORS	*Community Colleges; *Computer Assisted Instruction; Computer Graphics; Computer Programs; *Cost Effectiveness; Costs; Curriculum Development; *Demonstration Programs; English Instruction; Instructional Improvement; Instructional Innovation; *Instructional Systems; Instructional Television; Interaction; Junior Colleges; Mathematics Instruction; On Line Systems; Program Descriptions; Time Sharing
IDENTIFIERS	Digicolor Systems; Minicomputers; TICCIT; *Time Shared Interactive Computer Control Inform TV

ABSTRACT

The MITRE Corporation is disseminating computer-assisted instruction (CAI) through a demonstration program funded by the National Science Foundation. The goal is to show that CAI can provide improved, cost-effective instruction in community colleges. Significant products include: two demonstration systems, each with 128 student terminals; four semesters of math and English courses; and a complete package of authoring and delivery software. Some of the major innovations of the Time-Shared, Interactive, Computer-Controlled Information Television (TICCIT) system include: 1) courseware designed to produce mastery, improve learning strategies and student attitudes, and develop responsibility; 2) the use of audio and color television displays (the Digicolor System); 3) minicomputers to power self-contained systems; 4) low system cost of approximately $450,000 for one system or $250,000 for moderate quantities; 5) efficient courseware production procedures; 6) an on-line authoring system which produces quality CAI; 7) a learner control command language which fosters efficient strategies and positive attitudes; and 8) an implementation planning model which promotes faculty acceptance and system integration into colleges. (Author/LB)

M74-1

AN OVERVIEW OF THE TICCIT PROGRAM



January 1974

THE MITRE CORPORATION

M74-1

AN OVERVIEW OF THE TICCIT PROGRAM

January 1974

i/ii

THE MITRE CORPORATION
WASHINGTON OPERATIONS

ABSTRACT

The MITRE Corporation is attempting to catalyze the mass dissemination of an educationally sound form of Computer-Assisted Instruction (CAI) through a multi-year program of development, demonstration, and evaluation, sponsored by the National Science Foundation. The goal of this program is to demonstrate that CAI can provide today better instruction at less cost than traditional instruction in community colleges. Significant products of the program include: two demonstration systems, each with 128 student terminals; four full semesters of community college math and English courses replacing up to 20 percent of all classroom time taken by the average student, and a complete package of authoring and delivery software. Major innovations include:

- Courseware in math and English designed according to a new synthesis of instructional theorems to produce high levels of mastery, and to improve learning strategies, to foster attitudes that lead to approach rather than avoidance, and to develop responsibility.
- The use of audio and color TV displays (the Digicolor System) in the student terminals to provide voice-accompanied multicolored alphanumeric and graphic displays, as well as full-color sound movies.
- The use of a pair of minicomputers to provide the necessary computer power in a self-contained system of 128 terminals.
- A low system cost—roughly $450,000 today for one complete, self-contained system with terminals, and a projected cost of less than $250,000 in moderate quantities.
- Team organization and training procedures for cost-effective large-scale courseware production which is transferable to education or industry.
- A new on-line authoring system styled to support the production of high-quality CAI.
- The innovative use of a "learner control" command language to facilitate the development of efficient strategies and positive attitudes.
- An implementation planning model designed in cooperation with skilled college administrators and teachers which can facilitate faculty acceptance and system integration into colleges.
- A projected commercial cost including hardware, equipment maintenance, and CAI programs of less than one dollar per student contact hour.

This CAI system has been termed TICCIT, for Time-Shared, Interactive, Computer-Controlled Information Television.

ACKNOWLEDGMENTS

The National Science Foundation, under contract number NSF-C729, supports the program described in this publication.

The ideas, concepts, and designs reported herein are the work of many individuals. The TICCIT staff numbers over sixty people, including those of our major subcontractor for courseware, Brigham Young University. Credits for even those who gave direct contributions to this report are too numerous to properly attribute. Furthermore, close cooperation and consultation with numerous research, academic, and industrial organizations have been responsible for much of our progress and conclusions. However, it is necessary to note the significant contributions made to the TICCIT program by Kenneth Stetten, C. Victor Bunderson, Erik McWilliams, and Arthur Melmed. Ken developed the TICCIT design concept and, for the past three years, led the TICCIT program serving as its Principal Investigator. Vic Bunderson leads the courseware development effort at Brigham Young University, and has been personally responsible for countless improvements in the TICCIT system. Both Erik McWilliams of the National Science Foundation and Arthur Melmed, formerly at the National Science Foundation and now at the National Institute for Education provided a sophisticated level of review and guidance that encouraged needed refinements in many aspects of the TICCIT system concept.

Inquiries regarding the TICCIT program should be directed to The MITRE Corporation, 1820 Dolley Madison Boulevard, McLean, Virginia 22101; telephone (703) 893-3500.

John L. Volk,
Department Head
Computer Systems

William F. Mason,
Technical Director
Systems Development

Charles A. Zraket
Senior Vice President
Washington Operations

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

I. INTRODUCTION

Because of the difficulty of conveying all aspects of TICCIT in a short document, this paper is aimed at a technologically-oriented audience in industry or education. An additional perspective is provided by a companion document written by the TICCIT courseware and implementation design group at Brigham Young University. This document, "The TICCIT Project: Design Strategy for Educational Innovation," is oriented toward the perspective of a general audience of educators. It discusses the educational needs for TICCIT, the design goals, and the educational philosophy and strategy underlying the TICCIT courseware, software, and hardware design.

Computer-Assisted Instruction (CAI) has not been a commercial success. This has occurred despite instructional research that has demonstrated the effectiveness of CAI, and at a time when the problems of traditional instruction have never been more apparent. CAI systems have been offered by large companies and small, and school systems have never had larger budgets, but the dollars flow toward continuing support of traditional instruction. The commercial failure of CAI has been attributed to many factors: an initial oversell of its capabilities, an underestimation of the effort to produce high quality educational content, expensive, inappropriate and unreliable hardware, an educational bureaucracy resistant to the intrusion of computers in the classroom, and the decentralized structure of the American educational system that leads to tens of thousands of school systems each having to be individually sold on the idea. The CAI experience of commercial computer manufacturers in the mid and late 1960's was so bad that only a few continue to market systems.

MITRE's interest in CAI, begun four years ago, sprang out of traditional instruction's failure to teach reading in many elementary schools. During these past years, many proposals have been written, systems designed, and some of the enigmas of CAI understood. What became apparent was that neither a lower cost, higher performance CAI computer system nor an improved theory of instructional psychology would get CAI in the schools. The real problem is the making of a market (i.e., creating a supply/demand situation) for Computer-Assisted Instruction. We are attempting to sell the CAI concept to the schools as a reasonable, cost-effective approach to individualized instruction, while at the same time convincing the computer/textbook/publishing industries that CAI can be marketed now. To do this, we have selected the community college as a market area on which to focus our efforts.

The community college is particularly attractive as a place to start because of the rapid growth of community colleges, their receptivity to innovation and emphasis on educational effectiveness, the wide spread of student abilities leading to a dramatic need for individualized instruction, and, to a lesser degree, the trend toward statewide control or administration of community college systems. We have studied the needs of community colleges, selected courses with the highest exposure, and sized and priced our system to be suitable for most community colleges. By using generally off-the-shelf components to build our system and by developing courseware (the educational software) in a structured, project team environment, we are attempting to show industry that the hurdle of hardware development is already behind us, that excellent CAI courseware can be dependably developed by a team of professionals with complementary skills, and that an implementation/service relationship can be established with college faculties and administrators that paves the way for successful introduction of CAI.

Phoenix College--a campus in the Maricopa County Community College District, Phoenix, Arizona--and the Alexandria Campus of the Northern Virginia Community College will be the sites for demonstration and evaluation of the TICCIT system. These two colleges are similar to other community colleges in terms of their enrollment, curriculum, socio-economic and performance spread of students, as well as minority representation in their student bodies. At this time, detailed plans for the implementation of TICCIT in these colleges are being prepared.

This paper is divided into six parts: System Concept, Hardware, Courseware, Authoring System, Software, and Implementation Plan. The rationale for various features of the TICCIT system is reviewed in the System Concept section. The hardware is discussed, with emphasis on its functional aspects rather than technical detail. In discussing the educational content (courseware) being prepared for the system, particular attention is given to the modular structure of the courseware system and the learner control command language. The team-oriented authoring system is then explained with examples of how the TICCIT authoring forms are used. Next, with the basics of the hardware, courseware, and authoring system in hand, the TICCIT computer software is explained. The TICCIT application software and the operating systems, along with the results of a computerized model of the system's response characteristics, are discussed. The plans for integrating the TICCIT system into the community colleges are discussed in the paper's final section.

II. SYSTEM CONCEPT

MITRE sees the hardware, authoring, and delivery of software as subsystems in a larger system concept. A market success for CAI depends on the effectiveness of the courseware and the skillful design and execution of the implementation plan. The goal of the system is to serve students, faculty, and administrators. Unlike earlier CAI systems which have had to be designed around the marketing strategies of computer companies seeking to sell existing business and scientific data-processing equipment, MITRE has attempted to engineer a system around the elements that the courseware must contain to be effective and to consider the requirements of students, faculty, operational personnel, and administrators in an operational environment. All subsystems, hardware, software, courseware, and implementation procedures are designed to work together toward the goals of low cost, reliability, educational effectiveness, and informed acceptance and use by all classes of users.

The educational design strategies in this larger systems concept are not emphasized in this paper, but may be obtained in the companion document referred to above. By "systems concept" we refer here to some of the more salient features of the hardware configuration and authoring approach, to distinguish TICCIT from other CAI systems.

MITRE's TICCIT system is expected to be the first of a new generation of CAI systems to demonstrate instructional delivery in full cost-effective operation. It reflects much of what others in the past decade of CAI research have found useful, and includes many of the significant advances that have been made in the computer/semiconductor/communications industries. The TICCIT system is conservative. It is based on proven technology, and provides a careful selection of capabilities that foster educational richness while maintaining low system cost.

At any one time, the TICCIT system instructionally interacts with more than 100 students, each moving at his own pace through any of the system's CAI courses. The system retrieves logic, student records, audio, and visual information from its various data bases, and provides an audio-visual presentation for the student at his TV terminal. The student responds to the system by typing on a typewriter-like keyboard which invokes another computer-generated audio-visual presentation. Time-sharing techniques are employed in the system to allow simultaneous access to the data bases.

There are two distinct components of any CAI system: a system to allow assembling of the data bases (i.e., the conversion of the author's material into the computer data bases), and a system to deliver interactively the data bases to the students. In many developmental CAI systems, these two components have been combined into a single system. While this has proven to be useful where the authoring load is substantial, it can result in relatively inefficient delivery systems (primarily because of the use of interpreters instead of executing compiled code) and in systems that cannot be adapted to different authoring languages or approaches without reworking both the authoring system and the delivery system. CAI material is expensive. In the future, therefore, it will most likely be prepared by professionals, packaged, sold, and serviced in much the same manner as computer software, or complex audio-visual systems. The authoring component of the CAI system may be physically separated from the delivery components, and, in fact, the authoring component may be composed of entirely different hardware than the delivery system. For reasons of efficiency, better authoring, and the long-term approach to the generation of CAI, the TICCIT system design separates authoring and delivery. MITRE's system, as described in this paper, is optimized for the delivery of CAI; however, it does have an "on-line" authoring system.

The separation of content and strategy logic in the TICCIT courseware design gives the TICCIT authoring system a set of unique characteristics. No "CAI language" in the conventional sense is visible. A form-oriented authoring approach keeps teams of authors separated from coding and command language complexities. A team of packagers code the formatted manuscripts produced by the authors and enter and process the resulting content files. The packagers code in a form-oriented manner too, except when a special game or simulation is required. Then they use Algol or assembly language, putting all of the power of the computer at their disposal. Strategies and tactics are placed primarily in the hands of the students who are given a "learner control command language," and an "advisor program" to guide them in proficient use of this language.

TICCIT, as most other CAI systems, relies on still (not moving) computer-generated TV displays, featuring alphanumerics and graphics, for most instructional sequences. A wealth of knowledge of the effective use of this medium of instruction has been assembled, as well as of the general system requirements it demands. For instance, it is generally accepted that with this form of instruction, the computer generates a new display for each student on the average of once every 15 to 30 seconds; and, again on the average, the student responds with one key push every two seconds. As CAI assumes more of a role in everyday instruction, the need for variety will grow. The more tools the instructional designer has at his disposal, the greater his chances of succeeding in motivating, maintaining, and focusing the attention of his students.

The addition of audio messages to visual presentations will increase students' attention span and give needed changes of pace in extensive CAI courses. Short audio messages can focus students' attention, provide mnemonic aids, and pace learning. Audio can broaden the range of instruction for which CAI is an appropriate mode of instruction. In this program's English course, for example, the inclusion of audio will allow the author to demonstrate differences in dialect that will liven discussions of regionalisms and will also make testing of spelling skills possible. Music snatches and sound effects will liven the lessons and add effect.

The availability of color presentations on TICCIT terminals is another step toward enlivened CAI. Like the addition of sound, color will increase variety and serve as a motivational tool. Color has long been used in the television industry to add to the credibility and reality of a production. In instructional programs, it can also be used to differentiate and emphasize crucial material and to prompt student responses.

TICCIT's ability to display video tapes on an individual basis at student terminals under computer control represents a cost-effective means to project any kind of motion sequence with sound and color. It is MITRE's belief that the use of a mixture of video tape sequences with frame-by-frame CAI interactions is a more cost-effective approach to educating large numbers of students than attempting to store or generate all images from a central computer. By utilizing the great variety of presentation techniques developed in the motion picture and television industries, and by exploiting full-fidelity audio capability and color, the author of educational materials has far more variety at his disposal than if he were provided only with a graphic CRT terminal. Computer-graphic sequences or computer-generated animations as well as sequences with actors, language teachers, etc., that are effective can be mass disseminated through the use of video tapes. The time has come to expand the capabilities of CAI so that the expertise of educational television can be brought to bear on instructional problems.

The time has passed when the users in the educational market place need to make do with teletypes, character only CRT's, and other business machines awkwardly adapted for instruction. The capabilities of the TICCIT terminal, the low cost of the system, and the design of hardware, software, and courseware toward valid needs and goals of the intended users in education and training can change this picture dramatically. Courseware which is effective and appealing, system capabilities which are rich and exciting, and authoring software which is human-engineered for CAI can open new markets far beyond the uses in community colleges for which the original TICCIT systems are being developed.

III. HARDWARE

MITRE's TICCIT system (Time-Shared Interactive Computer Controlled Information Television) is composed of off-the-shelf hardware and a few specially designed subsystems. It uses color TV receivers the student terminal displays. Its pair of minicomputers (see Figure 1) can support over 100 active terminals. The system not only serves students in centrally located clusters of terminals, but can also serve students in dormitories, offices, and even homes off-campus through special terminals connected to the computer system via a standard cable television system.

TICCIT Terminal

The student terminal is the most important piece of hardware in the system. It must provide a pleasant and efficient interface between the student and the computer. From the student's viewpoint, the terminal is composed of a color TV display, a pair of headphones, and a keyboard. The terminal displays alphanumerics and graphics in seven colors, under computer control, as well as full-color videotapes. Up to 17 lines of 43 characters each may be displayed. The character set is completely programmable, with up to 512 distinct characters being definable at any one time.

Graphic displays are constructed on a bit by bit basis on a grid of 204 elements in the vertical direction by 430 elements in the horizontal direction. The color of each character may be individually specified. Short (about five-minute), full-color videotape add variety to the computer/student interaction and give the courseware writer a strong tool to dramatize difficult concepts. The student may view videotapes for about five minutes out of every hour.

To further augment the visual display, the pair of headphones brings prerecorded messages under computer control, to help a student with concepts with which he is having difficulty. It is expected that the student will be listening to audio responses (selected from a random-access audio data base containing about five cumulative hours of audio messages) only 10% of the time he is at the terminal and that audio responses will vary in duration from about one second to approximately 10 or 20 seconds. The hardware is implemented to support the level of use of audio and videotapes that the authors of our courseware feel is appropriate, but because of the modular nature of the audio and movie hardware, the hardware capability can be expanded or contracted as appropriate.

An electronic keyboard enables the students to talk back to the computer. While the keyboard is of a standard type, a special key layout was selected to simplify operations. As shown in Figure 2, the center of the keyboard is similar to that of a standard IBM Selectric typewriter. The keys on the left provide cursor and editing control, the keys on

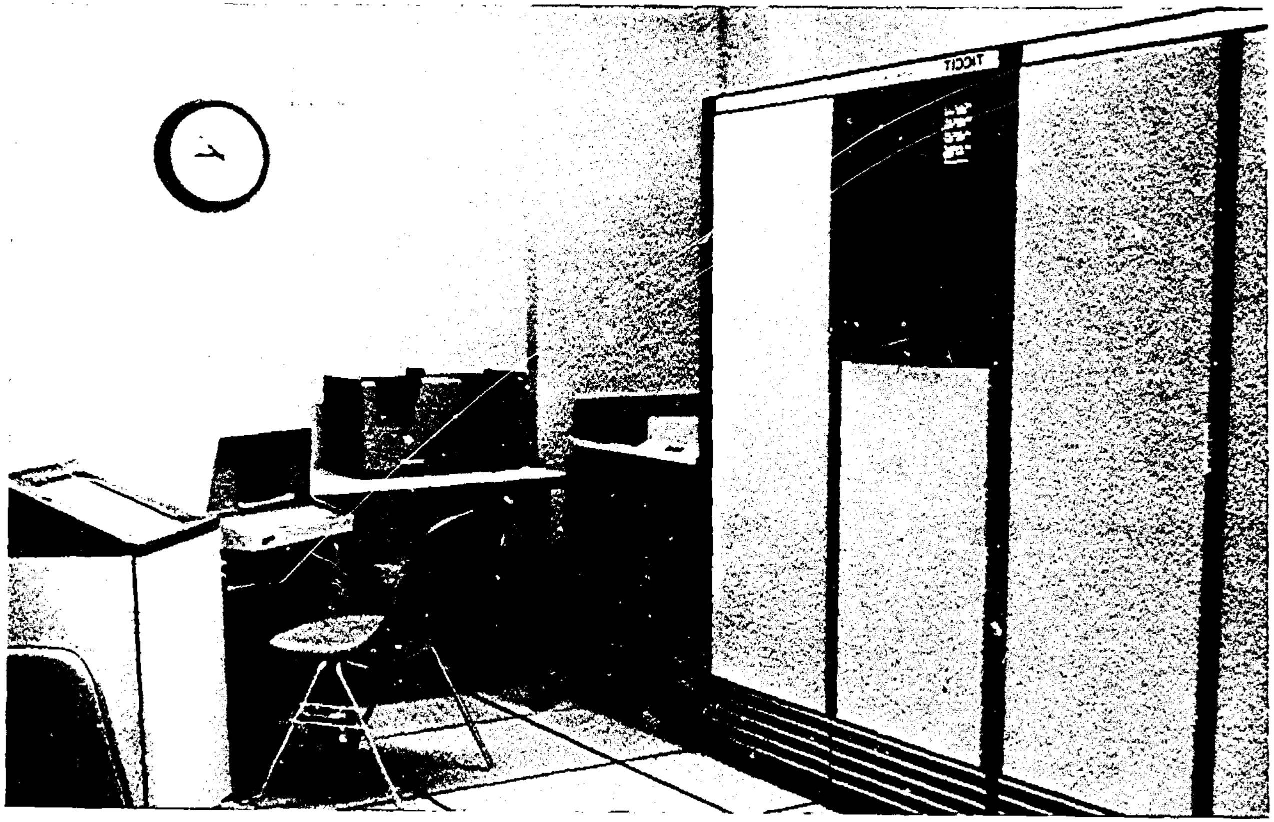


FIGURE 1
TICCIT COMPUTER SYSTEM

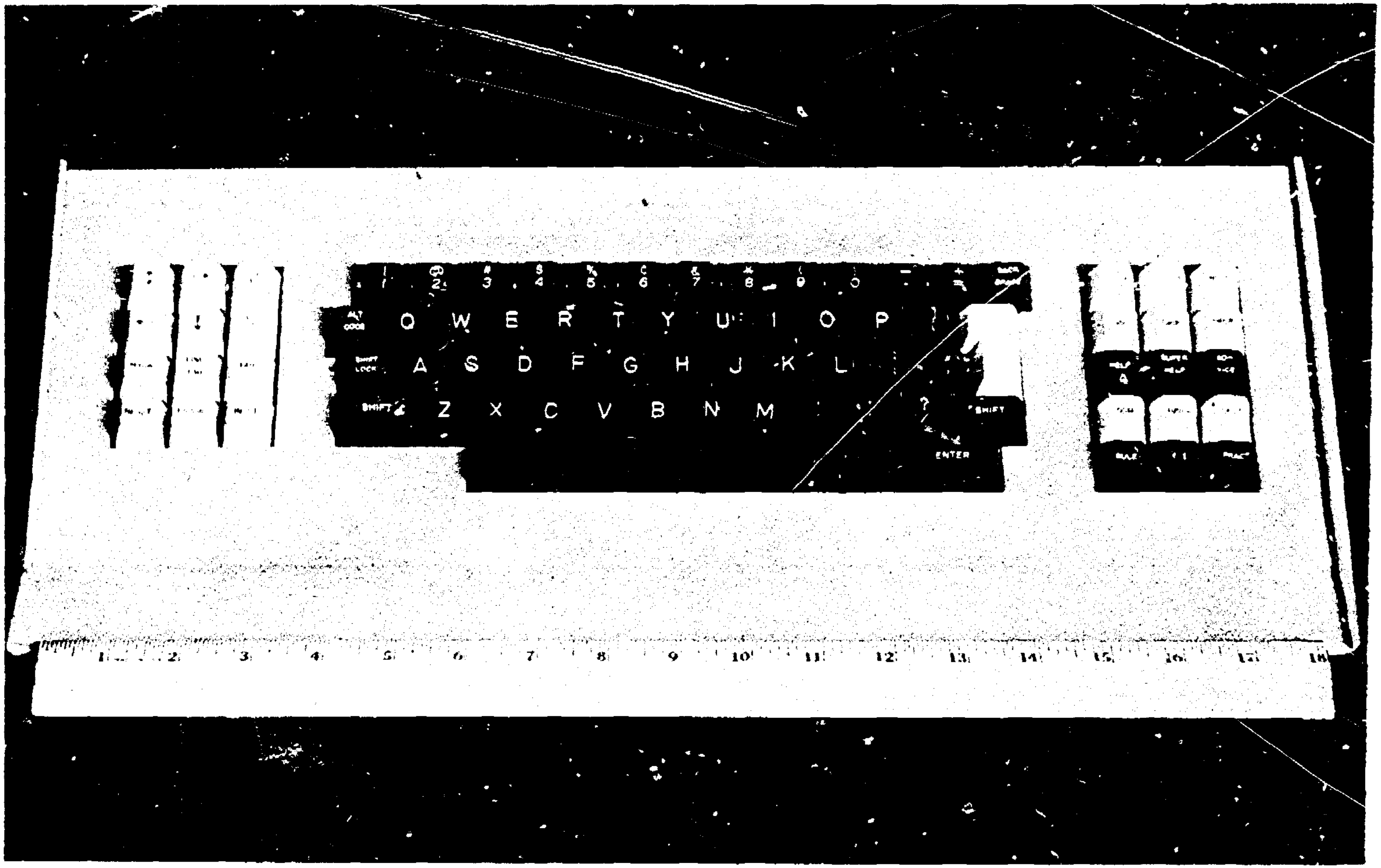


FIGURE 2
DETAILED VIEW OF STUDENT TERMINAL KEYBOARD

the right provide learner control options. MITRE considered other student response devices, such as the light pen, mouse, joystick, etc., to supplement the keyboard, but rejected them because non was available commercially at an acceptable cost and their instructional importance (at the community college level) was questionable.

TICCIT Computer System

A block diagram of the TICCIT computer system is shown in Figure 3. The system separates terminal and processing tasks with a minicomputer devoted to each task. The terminal processor performs all fast-reaction, highly-stereotyped functions, interacting with the TICCIT student terminals, including frame outputting as well as keyboard input multiplexing. The main processor, utilizing the TICCIT data base, generates and assembles frames to be displayed as a function of courseware and student responses. Tasks of the main processor are diverse and relatively slow-paced.

The main processor, a Data General Nova 800, is configured as a time-sharing minicomputer with 49,152 words of core storage, special hardware time-sharing protection features, and the usual host of standard peripherals, in addition to two large moving-head disc drives and a small fixed head disc. The peripherals (card reader, magnetic tape unit, line printer and CRT console terminal) are all low and medium speed, low-cost items ideally suited for courseware development and for later use in administrative applications.

The terminal processor, a Nova 800, is similar to the main processor, but with less memory. As its name implies, it services the TICCIT terminals by receiving and processing keyboard entries, controlling the audio response system, and by generating new displays to be sent to the terminal (for instance, "echoing" characters to the student terminal display as they are typed). The buffered computer-to-computer link uses a direct memory-to-memory data transfer system to provide fast data transfer.

The TICCIT courseware data base is split between two large moving head disc drives (IBM 2314 equivalent). The disc drive on the terminal processor stores predominately graphic material, while the data base disc on the main processor stores the educational text and answers processing instructions. These two disc drives are sufficiently large to hold the four full semester courses in their entirety. Additional disc drives may be added at a cost of $12,000 each to expand the data base. Another moving-head disc of the same type, connected to the main processor serves as the system swapping disc and holds student records for more than 3000 students. The other two disc drives on the terminal processor contain the audio data base.

The Digicolor System

Figure 4 shows how TV pictures are passed from the computer center to the TICCIT terminal in the Digicolor System. Under computer control a single display

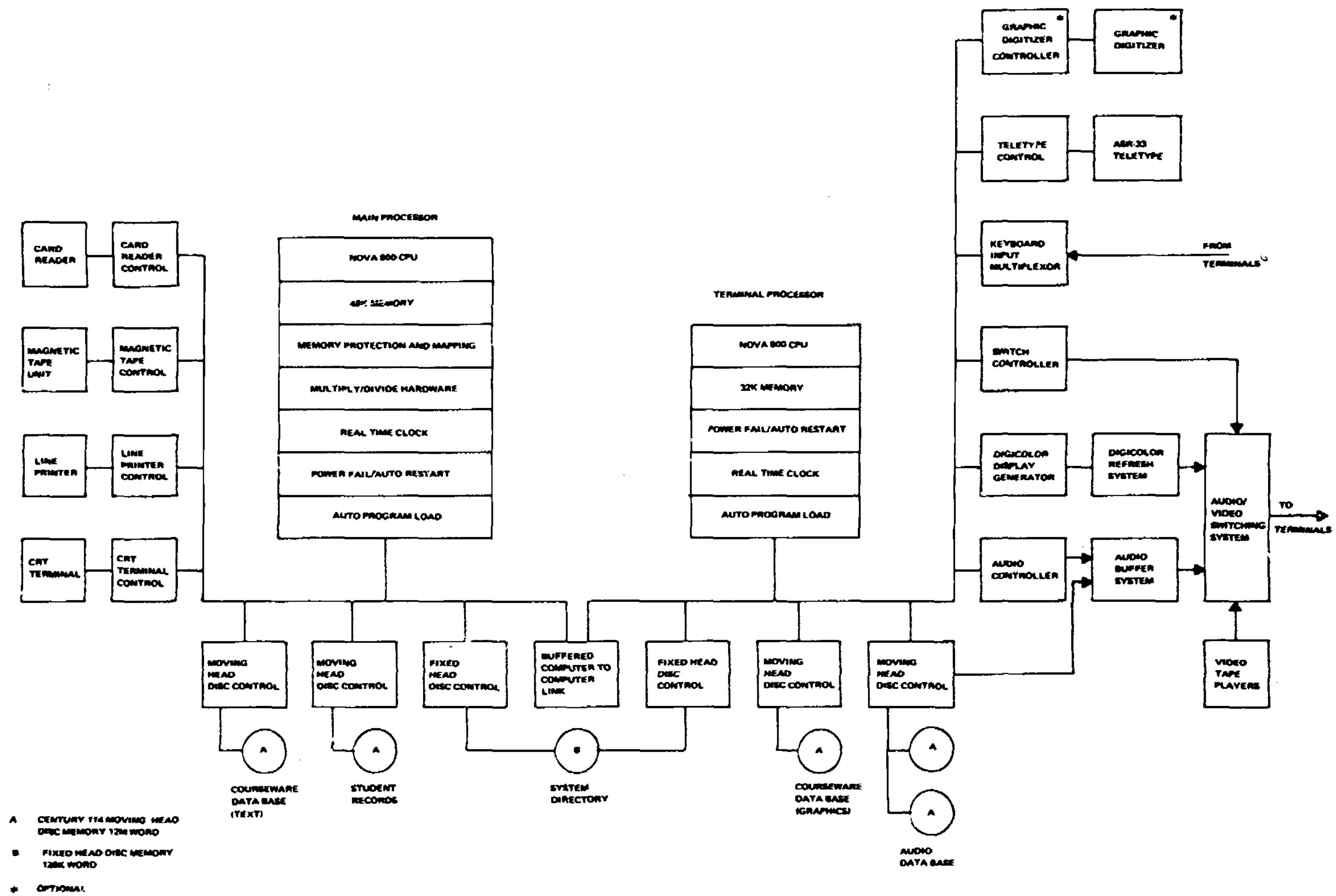


FIGURE 3
TICCIT COMPUTER SYSTEM

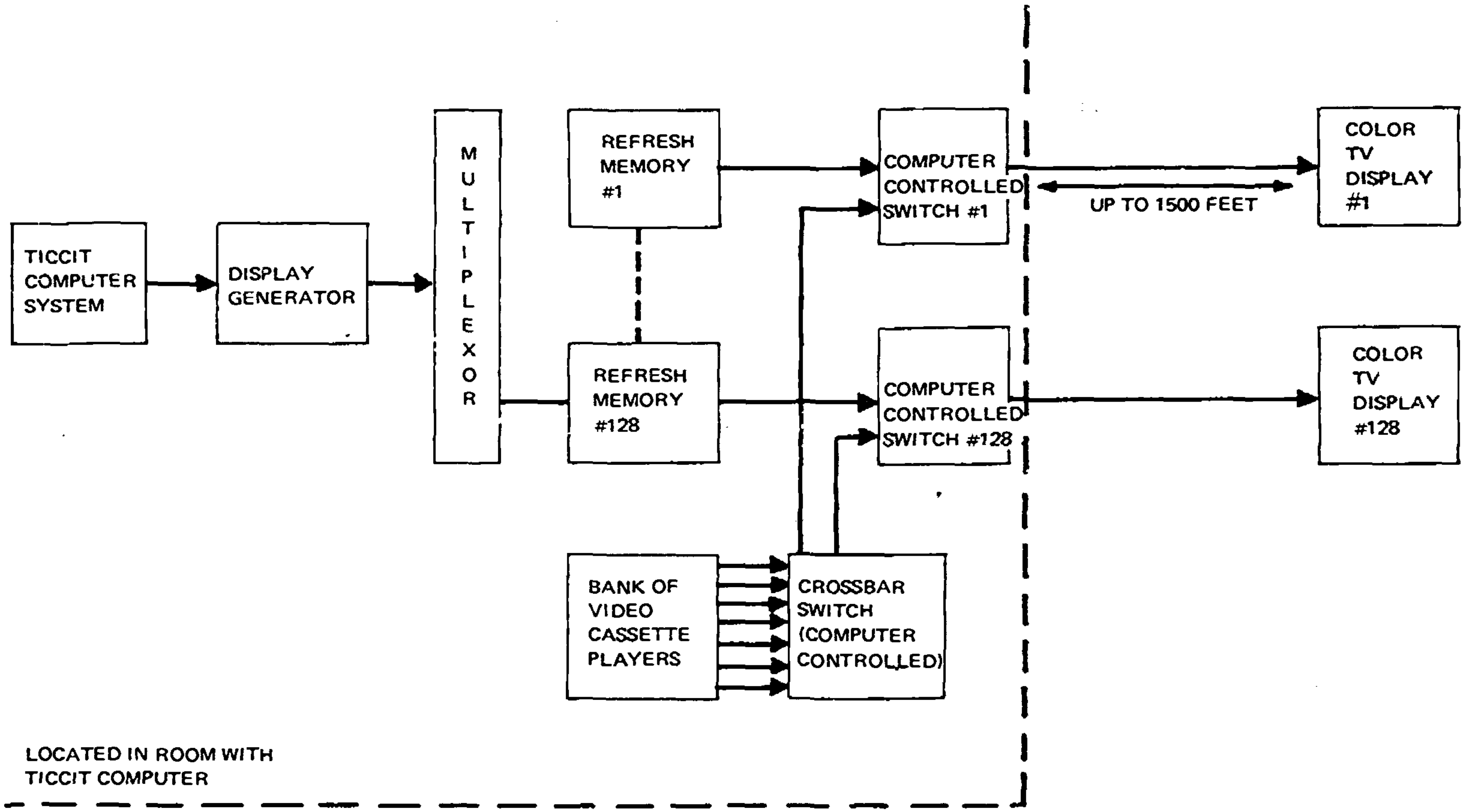


FIGURE 4
SIMPLIFIED BLOCK DIAGRAM OF TV PICTURE GENERATION AND TRANSMISSION

generator is time-shared by all terminals. Its output (typically a single frame TV picture) is selectively passed to the video-refresh memory of the appropriate student terminal. The refresh memory repetitiously sends a single TV picture, originally generated by the display generator, to its associated color TV receiver.

The refresh memory is composed of two MOS memory subsystems. One memory, which contains luminance information for the display, is an 88,000-bit dynamic shift register, operating at about a 10MHz rate. The color information is contained in a MOS random-access memory, organized as three bits by 1,500 words. The luminance refresh contains the bit-by-bit detail of the display, while the color refresh holds the color on essentially a character-by-character basis. (In fact, color information is stored separately for the top half and bottom half of each character to allow more flexibility with colored line drawings.) The luminance refresh is quite small, requiring only a single 8" x 10" printed circuit board; the color refresh is much smaller—four such units are mounted on a similar-sized board. The benefits of this digital solid-state refresh over earlier refresh techniques (such as electronic storage tubes) are greater reliability, simplified maintenance (no analog adjustments), resolution two to three times better, and comparable cost.

A bank of 20 video cassette tape players (computer-directed but manually operated in the demonstration systems) provides the source of full-color movies. Each TV receiver is switched to receive video either from its video-refresh memory or a selected videotape player. A computer-controlled crossbar switch is used in this interconnection process.

The terminals may be located up to 1,000 feet from the computer center. Audio information being sent to the terminal and keyboard signals coming to the computer are carried on separate twisted pairs in the same multiconductor cable that carries the video information to the terminal.

Because the TICCIT terminal display is a television receiver and requires a signal similar and compatible with that of normal television, a cable TV system can carry TICCIT signals. Several techniques to deliver CAI to the home via a cable television system have been developed and are being studied in a complementary program at MITRE.*

TICCIT Audio System

The TICCIT audio system delivers prerecorded audio messages to the student terminals. It can randomly access over five hours of digitally recorded sentences and phrases, and provide audio to as many as twenty students simultaneously. It is expected, though, that on the average, students will be receiving audio ten percent of the

*"Some Technical, Economic and Applications Considerations of Interactive Television," Charles A. Zraket, MITRE publication M73-40.

time they are receiving instruction; which means that only twelve students will, on the average, be simultaneously receiving audio. The capability of serving twenty simultaneous students allows for the peaking factor and guarantees that it will be rare when all audio lines are in service and a student must wait for a free audio line.

The recorded audio is converted to a digital form for storage with a digitizing technique known as adaptive delta modulation. With this technique, input speech is sampled by an analog-to-digital converter 20,000 times per second. One bit is used to represent each sample. For a given sample, a 1 is stored if the energy in the input speech is greater than a reference level. A 0 (zero) is stored if the energy in the input speech is less than a reference level. The reference level is then adjusted up or down by a predetermined amount so that it can be used for the next sample. The amount that the reference level is adjusted up or down can be made dependent upon what has happened in the last few samples. This makes the technique adaptive. A characteristic of the speech waveform is that it changes in amplitude and direction quite rapidly. The adaptive approach aids in tracking a rapidly changing waveform.

Figure 5 illustrates this approach. The reference level (which represents the output signal) is initially set to represent silence. Two parameters are used in changing it. One is the minimum single step change in reference level and the other is the maximum single step change in reference level where the maximum is a multiple of the minimum. If the first sample of speech is greater than the initial reference level, then a 1 is stored and the reference level is increased by the minimum single step. If the second sample is greater than the current reference level, then a 1 is stored and the reference level is increased by twice the minimum step size. For the third increase in a row, the reference level is increased by four times the minimum single step. For the fourth increase in a row, eight times the minimum single step change is used. This is the maximum single step change. When the speech sample changes direction (i.e., in this case, is less than the reference signal), a 0 is stored for the sample and the reference level is decreased by an amount equal to one-half of the previous multiplier times the minimum step size (i.e., in this case, 1/2 of 8 or 4 times). This adaptive approach works well with rapidly changing signals, yet damps out quickly if the signal tends to be steady. The adaptive delta modulation technique is not new. It was implemented on a speech generation system by The MITRE Corporation in 1969.

The overall configuration of the audio system is shown in Figure 6. An analog-to-digital and a digital-to-analog converter are attached to the terminal processor. Speech comes in through the microphone, is digitized, and initially stored in the terminal processor memory. It can immediately be sent through the D-A converter to a speaker so that an operator can listen to the quality of the speech. The operator, through appropriate software, can remove unwanted portions of the recorded speech and listen to the effect. After

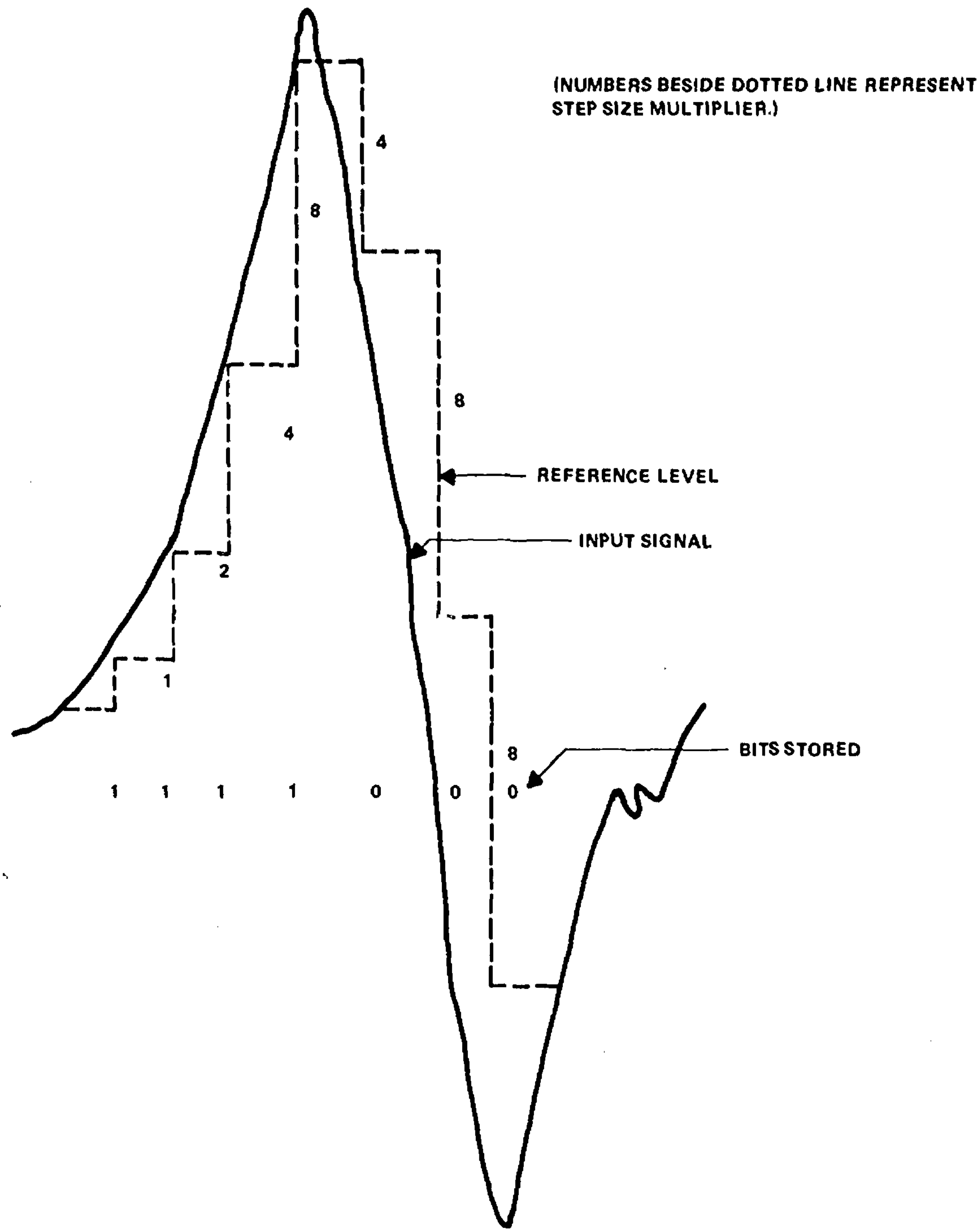


FIGURE 5
ADAPTIVE DELTA MODULATION TECHNIQUE

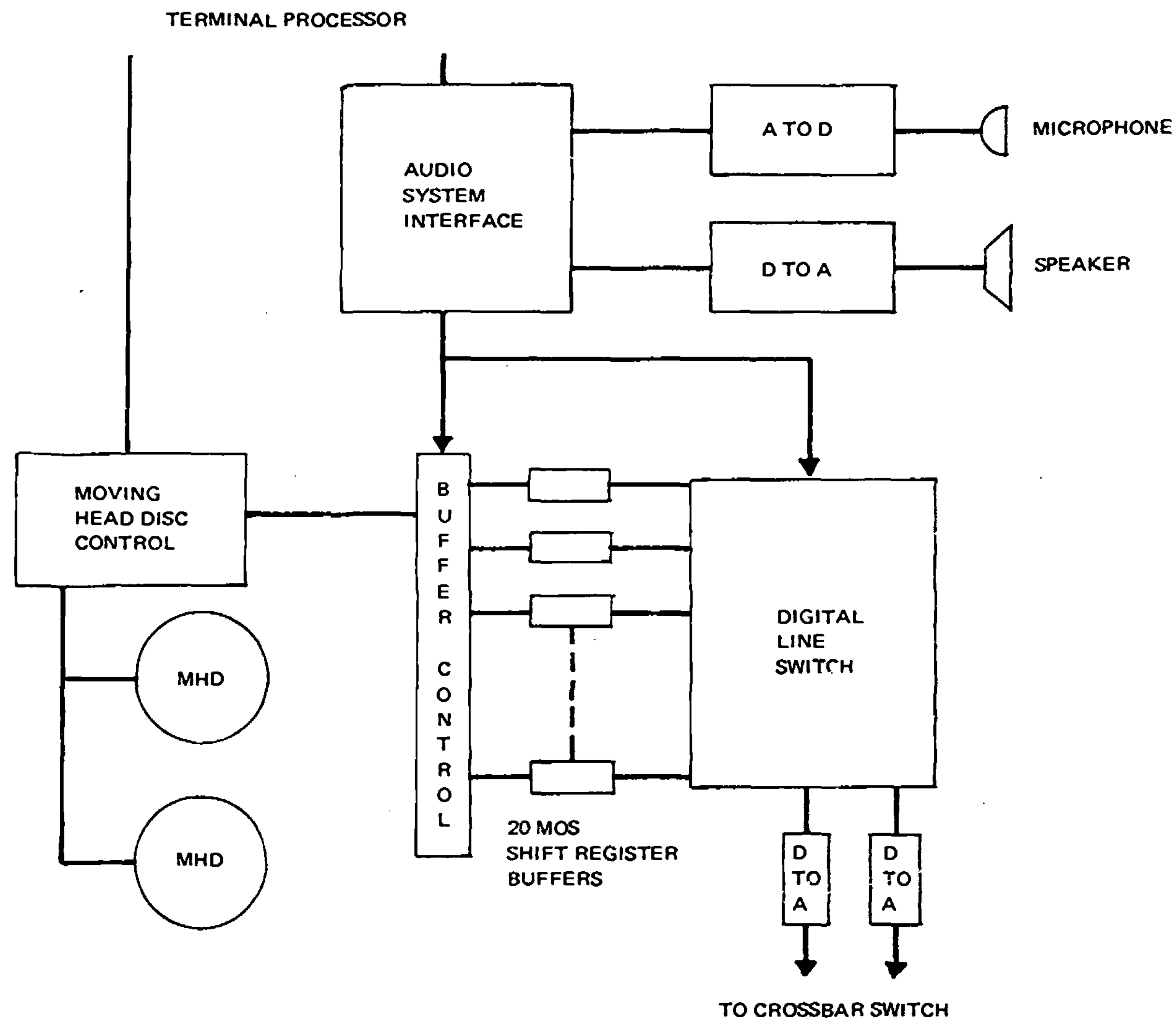


FIGURE 6
TICCIT AUDIO SYSTEM

the operator has the speech sounding the way he wants it, he will give the audio message a name and store it on one of the two disc drives dedicated to audio. The name, length, and location information is stored in a fast access dictionary on one of the discs.

In the delivery mode the system operates in the following manner. A message comes from the main processor to the terminal processor with the name of an audio message. In the terminal processor the name is looked up in the fast access dictionary and its location is found. This location information is used to position the read head on the appropriate disc. The terminal processor next gives a command to the buffer controller, which causes the audio data to be read from the disc to one of twenty MOS shift register buffers. Audio data does not go through the core memory of the terminal processor, resulting in minimal load on the terminal processor for audio delivery. The twenty MOS buffers and associated circuits perform the function of speed matching. Data is unloaded from disc at high speed (2.5 million bits per second) and output to the digital-to-analog converters at 20,000 bits per second. From the MOS buffers the data is sent through a line switch and then to the digital-to-analog converters. The digital line switch allows the output of any MOS buffer to be connected to any one of the twenty audio output lines under control of the terminal processor. The lines from the D-A converters go to the crossbar switch (part of the same crossbar switch that connects videotape players to the student terminals), which allows the lines to be connected to any one of the 128 student terminals.

Graphic Digitizer System

The design of the TICCIT system software permits the use of graphics (art work, drawings, cartoons, sketches, etc.) as an integral part of courseware displays. Due to the large volume of graphic usage (over 10,000 graphics are used in the four courses under development), graphic production is a task best performed by artists off-line. Therefore, a requirement existed to allow hand-drawn graphics to be automatically digitized for display by the TICCIT system. A graphic digitizer (Figure 7) was developed which permits semi-automatic entry of graphics into the TICCIT computer system.

The graphics are drawn using colored pens on specially prepared paper. The graphic paper has a grid that represents the character spaces in a TICCIT frame. This grid facilitates proper placement of graphics within a TICCIT display. An operator aligns reference marks on the paper with electronically generated reference marks on a TICCIT terminal. Then he places the digitizer in automatic mode.

Input is now initiated under computer control. The paper is scanned by a modified, high-resolution television camera with special circuits added to obtain uniform resolution and shading over the whole face of the vidicon, permitting color separation by

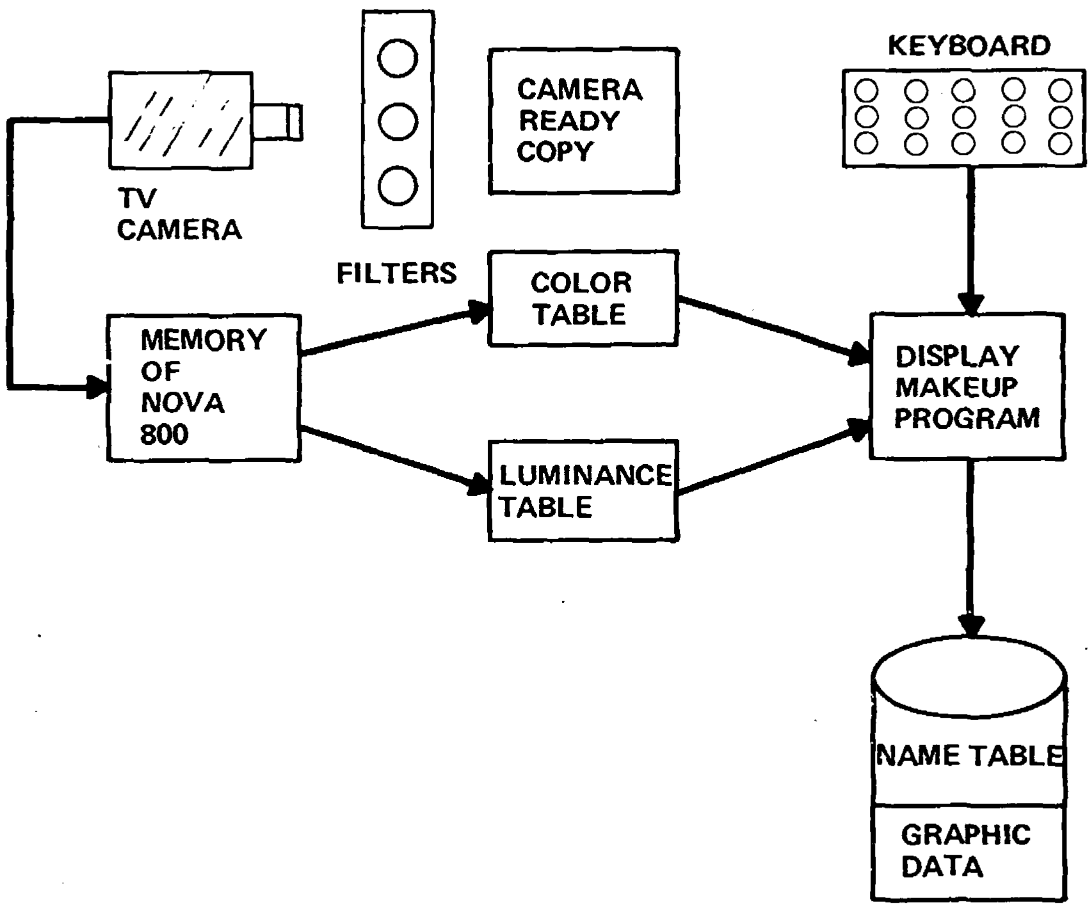


FIGURE 7
SEQUENCE FOR STATIC GRAPHICS

level slicing after automatic normalization. Using the clocking signals developed for the MOS luminance refresh, the scanned luminance data are automatically normalized for the TICCIT frame and stored as 87,720 consecutive bits in computer memory. The input is scanned three times in this manner through three colored filters (red, green, and blue) to provide recognition of up to six colors: red, green, yellow, blue, magenta, and black.

Computer software processes the color-separated luminance information to keep only significant information and to determine actual picture element luminance and half-character color. A special color "averaging" algorithm is used to assure that input noise will not affect the desired color specifications. From this information, a TICCIT display frame is created by partitioning the data into character definitions, defining the required characters, and displaying these characters with the necessary color information. This frame is then displayed on the TICCIT terminal.

If the operator judges the resulting display to be an adequate replica of the graphics input, he requests that it be stored on a disc for future reference and specifies the associated TICCIT graphic label as an identifier. At some later time, this graphic may be edited, if necessary, using the graphic source file editing program.

Cost of TICCIT

The cost of the TICCIT hardware today is shown in Table I. Excluding system integration and installation costs, but including the cost of the computer and all required hardware for audio, color, and video-tape capabilities, the prorated cost amounts to $3,600 per terminal. This is already a very attractive figure, but if the trend in cost reduction in minicomputers, moving-head disc drives, and integrated circuits continues, and even if only 25 systems were constructed in 1975, the total cost of the system would fall to $2000 or less per terminal. Less than 25% of the TICCIT system's cost is in the student's terminal hardware. Today, the equipment in the carrel costs $500 with a projected drop to $400.

Reliability and Maintainability

Special effort has been made on the TICCIT system to reduce both the possibility of system failure and the effort required to quickly repair the system if something does go wrong. In general, this goal is being reached by using proven products, solid-state digital technology, modular implementation, and a central location of most system elements.

The utilization of "off-the-shelf" proven products (such as the TICCIT computer and its peripherals) results in a more reliable system because of the countless hours of tests and refinement that only is practical on mass produced equipment. Maintainability of the proven product is enhanced by the low cost of spare subassemblies and high quality technical documentation. The predominant use of solid-state digital technology in the TICCIT system, for example in the refresh memory, reduces the number of circuits that can be affected by components aging or temperature extremes. In addition, it is usually easier to find faulty components with digital circuitry than with analog circuits. The modular implementation of TICCIT (i.e., the computer's 17 identical memory modules, the 128 identical refresh memories, etc.) speeds maintenance by allowing sparing of parts on a complete module basis. Maintenance is also simplified by the central location of all complicated TICCIT sub-assemblies. When something does fail, it is expected to be at the central location, where the field engineer will have easy access to both the spare modules and test equipment.

Table I
TICCIT HARDWARE COSTS TODAY

MAIN PROCESSOR	$ 34,000
TERMINAL PROCESSOR	15,000
CARD READER	4,000
LINE PRINTER	11,000
MAGNETIC TAPE UNIT	9,000
MOVING HEAD DISC CONTROL (3)	25,500
MOVING HEAD DISC DRIVES (3)	36,000
FIXED HEAD DISC CONTROL	3,500
FIXED HEAD DISC DRIVE	5,000
CRT TERMINAL	3,000
COMPUTER-TO-COMPUTER LINK	3,000
CHARACTER GENERATOR	7,000
KEYBOARD INTERFACE	6,000
AUDIO RESPONSE SUBSYSTEM	56,000
TV SETS (COLOR) (128)	32,000
KEYBOARDS (128)	22,000
LUMINANCE REFRESH (128)	82,800
COLOR REFRESH UNITS (128)	11,000
SIGNAL PROCESSING AMPLIFIERS (128) & CABLE	16,000
VIDEO TAPE PLAYERS (20)	17,000
REFRESH CONTROL ELECTRONICS	9,000
TV MODIFICATIONS	9,000
CROSSBAR SWITCH	17,000
CABINETS	7,500
TOTAL	$441,300

We anticipate that a single, on-site field engineer will be required to minimize the possibility that any subsystem failure will result in the entire system being down for several hours. Because of the system's modularity, the structure of the system's software, and the spare parts planned to be stocked on site, it is expected that following most any failure, the system can be made operational (perhaps with a reduced capability, i.e. fewer students on line) in less than thirty minutes. The cost of maintenance includes a non-recurring investment of $20,000 in spare parts and test equipment, and an annual expense of about $20,000 for the field engineer's salary and replenishment of the spare parts stock.

IV. COURSEWARE

The educational material (courseware) planned for use in the demonstration is being developed (under subcontract to MITRE) by teams of instructional psychologists, subject matter specialists (individuals experienced in teaching the subjects), media specialists, and programmers at Brigham Young University under the direction of Dr. Victor Bunderson. Dr. Bunderson (formerly of the University of Texas) and his colleagues have been instrumental, not only in the development of the courseware, but, in addition, have been greatly influential in the development of the system hardware and software.

The courseware system's design, and the team procedures which are geared to the production of the courseware components, differ in substantial ways from earlier models for CAI materials. These differences provide the potential that the TICCIT courseware may represent an intellectual as well as a practical contribution. It is an intellectual innovation because the courseware is designed around instructional theorems having sound empirical footings where possible, and theoretical integrity elsewhere. The inventions of the primary instruction logic, the map displays, the advisor program, and numerous other aspects of the courseware were based on principles of sound man-machine communication. The courseware is controlled by the student by means of a high level command language—a characteristic of man-machine symbiosis—rather than administered under computer-control step-by-step after the analogue of teaching machines or programmed instruction. The courseware is a practical contribution because the modular separation of content and logic leads to rapid improvability, transplantability to other systems, differentiated staffing for cost-effective production, and quality control. The practical advantages of design toward appeal for human users of the system, as well as effectiveness and efficiency, can also be of great significance toward the goal of catalyzing a new market.

The courseware design began with an analysis of the institutional needs of community colleges and the individual needs of their students. Traditional instruction's problems of increasing costs, low or uncertain effectiveness, and associated student discontent were immediately apparent. With open enrollment, more and more students in community colleges were seen to lack high school mathematics, English grammar, and composition skills.

Selected Courses

The result of the needs analysis was the selection of courses in English and mathematics. These subjects have the first and second greatest enrollment figures in community colleges. It was found that most members of the student population needed work in high school-level algebra and English composition and grammar. In selecting and shaping the content for the courses (each approximately semester-sized), no attempt was made to innovate drastically. The mathematics courses are quite standard in content. The English course, however, required some novel approaches to adopt this subject to computer administration, and to structure the content toward a generative, consistently defined model of the process of producing adequate writing at the beginning level. In all courses, strong efforts were made to clarify the objectives and structure, and through modularity of design, to provide flexibility for a wide range of implementation patterns.

The TICCIT Algebra course is designed to provide a general review of basic arithmetic skills and instruction in traditional beginning and intermediate algebra. Also it includes a treatment of common logarithms, systems of linear equations, permutations, combinations, arithmetic progressions and geometric progressions. The topics are arranged to provide a modular curriculum selection. Together with the Elementary Functions course, it provides a modular structure for pre-calculus mathematics. A concerted effort is made to emphasize statement problems and to illustrate applications.

The Elementary Functions course is designed to provide a treatment of the function concept with extension to elementary functions including the polynomial, trigonometric, exponential and logarithmic functions with a brief coverage of the conic sections in coordinate geometry. It requires as a prerequisite successful completion of the traditional Algebra II course or the TICCIT Algebra course. Emphasis is placed on intuitive understanding of the concepts involved, with prudent and sparing use of rigor. Graphic interpretation of problems is used whenever possible; pains are taken to help the learner acquire a "feel" for and a common sense attitude toward the material and its place in the subject matter.

English Composition is the first quarter or semester of the traditional composition course. It offers maximum student practice with immediate feedback on many aspects of the student's writing. This course is a writing laboratory in which the computer handles specific instruction in grammar, mechanics, diction, sentence structure, and paragraph development. Assessment and evaluation of writing skill are left to a trained instructor. The one objective of the course is to enable the student to write clear and otherwise effective prose in general standard English.

Our needs analysis indicates that these courses can account for approximately 20 percent of the total student contact hours of instruction in the average community college. Each course covers about four semester hours of traditional community college curricula, although if the redundancy in content is taken into account, up to 7 math courses and 3 English courses may be replaced. It is expected that the average student will require about 40–50 hours of terminal time to complete a course.

Mainline Versus Adjunct

In approaching the development of the courseware, it is important to recognize the distinction between the two classes of CAI programs as illustrated in Figure 8. The

ADJUNCT	MAINLINE
--- CONTEXT PROVIDED BY TEACHER.	--- REDESIGN OF A COMPLETE INSTRUCTIONAL SYSTEM, INCLUDING THE TEACHER'S ROLE.
--- PROGRAMMING BY TEACHER AND STUDENT.	--- SPECIFICATIONS AND PROGRAMMING BY DESIGN-PRODUCTION TEAMS.
--- FITS WITH STANDARD CREDIT-HOUR SCHEDULING.	--- REQUIRES SELF-PACED SCHEDULING AND GRADING.

SOME CONSEQUENCES OF EACH APPROACH

ADJUNCT	MAINLINE
--- REPRESENTS AN ADD-ON COST.	--- GREAT ECONOMIC POTENTIAL: SUPPLANTIVE.
--- REQUIRES LOW TO MODERATE CAPITAL INVESTMENT.	--- REQUIRES HIGH CAPITAL INVESTMENT.
--- INCREASED EFFECTIVENESS: OPPORTUNITY FOR RESTRUCTURING OBJECTIVES AND SUBJECT MATTER.	--- INCREASED EFFECTIVENESS AND EFFICIENCY.
--- MODEST BUT VARIABLE SYSTEM REQUIREMENTS; USE SCIENTIFIC OR BUSINESS ORIENTED COMPUTER SYSTEMS.	--- SPECIFIC ENGINEERING DESIGN FOR EDUCATION.

FIGURE 8
A DISTINCTION BETWEEN TWO CLASSES OF CAI PROGRAMS

main difference between them is the role of the teacher. In the adjunctive approach, the teacher is the central figure, and the programs are supplements to traditional classroom and laboratory work or serve as a new kind of homework. These uses include problem solving, using the computer for simulation and modeling, drill and practice supplements or remedial units, and use of programs for illustrations during lectures or laboratories. It is the teacher or his students who do the programming in these instances. The products of this "cottage-shop" method of development are heavily dependent on the originator to provide the context; thus, dissemination is difficult.

The mainline approach being taken in this program, in contrast, is designed from the first for mass dissemination. A complete instructional system is redesigned for a substantial block of material, so that the role of teachers is redefined,* and eventually reduced in basic courses (primarily by relieving the teacher of class presentations) as the system becomes more technologically intensive and less labor intensive. Lock-step scheduling is replaced by a self-paced, individualized scheduling system, with a criterion referenced standard for grading to replace grading "by the curve." A design and development team, having total capabilities not often possessed by the individual teacher, is responsible for the courseware development, documentation, and packaging for distribution.

We expect as a consequence of the use of mainline CAI that the reduction of instructional labor costs, through a reduction in total teaching staff in certain courses and the use of lower cost paraprofessional proctors, will offset the cost of the TICCIT system, including hardware, equipment maintenance, and courseware. The cost difference between traditional and computer-assisted instruction is complicated by several factors. It is extremely difficult to establish a true cost-per-student contact hour for any educational institution. There are many ways--at many different costs--by which a TICCIT system could be procured and maintained by an educational institution. There are many ways courseware could be prepared and its cost prorated. A major task of the Educational Testing Service in its role as evaluator in this program is objectively to gather sufficient data to estimate the cost differential between CAI and traditional instruction in addition to determining the differences in educational effectiveness. MITRE has, of course, analyzed the operational cost of TICCIT and projects a commercial (leased) cost of less than one dollar per terminal-use hour, including hardware, equipment maintenance, and prorated courseware cost.

*The new role is seen largely as tutor-advisor/diagnostician and problem solver for individual students. The master teacher role is emphasized in courses which follow the basic TICCIT courses, or concurrently with TICCIT. All of the intellectual excitement, the projects, and the human-to-human interchange of imponderable values of education can take place better when students are better and more uniformly prepared, and when the computer has helped define what is not essentially a necessary product of human interaction.

The cost of courseware development appears quite high—approximately $1.5 million to develop the four courses in this program (this figure includes nonrecurring R&D costs for first-time courseware). If even moderate dissemination is assumed, the cost on a prorated student contact hour basis can be shown to be quite reasonable. For example, if only two percent of the nation's 1000 community colleges procured TICCIT systems, and used this program's four courses for a five-year period, the prorated cost per contact hour for courseware alone would be about 15 cents (based on 20 schools, 1000 hours utilization/year, 100 terminals, and 5 years, giving 10 million student contact hours). We expect that the effort required for courseware development will be lowered as experience is gained in its production and that substantially more than two percent of the schools will adopt CAI. Together, this should result in nickel-per-hour courseware.

The Separation of Strategy from Content

The courseware developed by the design and development team follows the concept that it is possible to view the way subjects are taught independently from what is taught. This fundamental principle of "instructional science" has led to the specification of guidelines and procedures for the development of course content and to the adoption of learner-controlled courseware. TICCIT's approach to learner control is based on a multilevel model of student computer communication and a taxonomy of instructional variables.

A model for student-machine communication developed by Pask (1969)* provides inspiration for the three levels of discourse implemented in TICCIT. Pask asserts that all communication between student and computer can be described as taking place in one or more special languages. The flow of instructional information sequenced according to fixed algorithms within the computer, and the answers to questions and problems entered by the student comprise what Pask calls the L^0 language. Discussion about the instructional process itself, and attempts by the student to control the process in some way, take place in L^1. It is possible also to define an L^2 language in which control processes can be discussed and modified.

In the TICCIT system, we speak of progressively higher levels of discourse, analogous to Pask's languages. These are:

Level 0- Computer presents information and asks questions; Student makes pacing responses and answers questions.

*Pask, Gordon. "Strategy, Competence and Conversation as Determinants of Learning." Progressive Learning and Educational Technology (Vol. 6, October, 1969)

Level 1 - Level 0 plus capability for student to command computer relative to strategy (defined primarily as sequence within a well-structured set of instructional components).

Level 2 - Levels 0 and 1 plus ability of student and computer to converse (in a highly structured way) about high-level concerns (strategy, attitudes, motivating contingencies, etc.). The student can take action to modify his processes of controlling the system, using control keys and commands.

Level 0 is implemented primarily within files of instances (examples and non-examples of concepts and rule usage) where students may look at worked examples, or may practice. This is analogous to the more conventional "tutorial CAI."

Level 1 is implemented as a "learner-control command language" using a special keyboard and a few control words.

Level 2 is implemented by an Advisor program, which refers to a set of student historical data (monitor) and communicates through "status displays" at course, unit, lesson, and segment levels, and through audio and visual displays.

The implementation of levels 1 and 2 make it possible to seek student growth toward the goals of improved strategies, approach rather than avoidance, and responsibility. The strategy for achieving these goals, in addition to the more conventional goals of mastery and efficiency, is described in other publications.

The learning components used within level 0 are constructed according to Merrill's taxonomy of instructional variables.* Merrill's taxonomy involves three classes of variables:

1. Presentation form

	Expository	Inquisitory
Generalities	Rule	
Instances	Example	Practice

The system deals primarily with concept learning and rule learning, so a generality is a definition of a concept, or a rule. An instance is an example or non-example of a concept or a rule in use.

*Merrill, M. D., and Boutwell, R., "Instructional Development: Methodology and Research." In (Ed.) Review of Research in Education, AERA, 1973. (Also Instructional Research and Development Department, Working Paper Series #33, Brigham Young University, Provo, Utah. $2.50)

Expository means to tell, inquisitory to ask. Inquisitory generalities (e.g., "define a concept") are rarely used, both because we do not wish to stress rule memorization, and because of the computer's limit in processing natural language.

2. Inter-display relationships

   This category involves the matching and pairing of examples and non-examples, the difficulty levels of instances, the scope of the generalities and instances, and their abstractness-concreteness.

3. Mathemagenic Information*

   This category involves prompting and cuing and other attention-focusing techniques. Specific techniques include attribute isolation, search strategies, mnemonic aids, and production strategies.

With this set of variables any instructional sequence in complex cognitive learning can be closely described.

Courseware Structure

The courseware structure is based on the assumption that the great majority of instruction, particularly at the community college level, can be represented as concept learning (classification behavior) and rule using. By the way we define classification behavior and rule-using behavior a corollary is that the structure of instructional material analyzed into these categories is hierarchical. The TICCIT courseware is hierarchically organized in four levels as shown in Figure 9.

At the course and unit level the student is exposed to introductory material, unit and lesson objectives, and comprehensive tests on the material presented at the lesson and unit level. The introductory material consists of videotapes and "minilessons."

At the lesson level the student is provided with an introduction to the lesson material, the objectives of each of the lesson segments, educational games related to the lesson's objectives, a test covering the lesson objectives, and some additional related material with a special mastery test for students desiring an A or B grade.

*Information which gives birth (gen-) to learning (mathema).

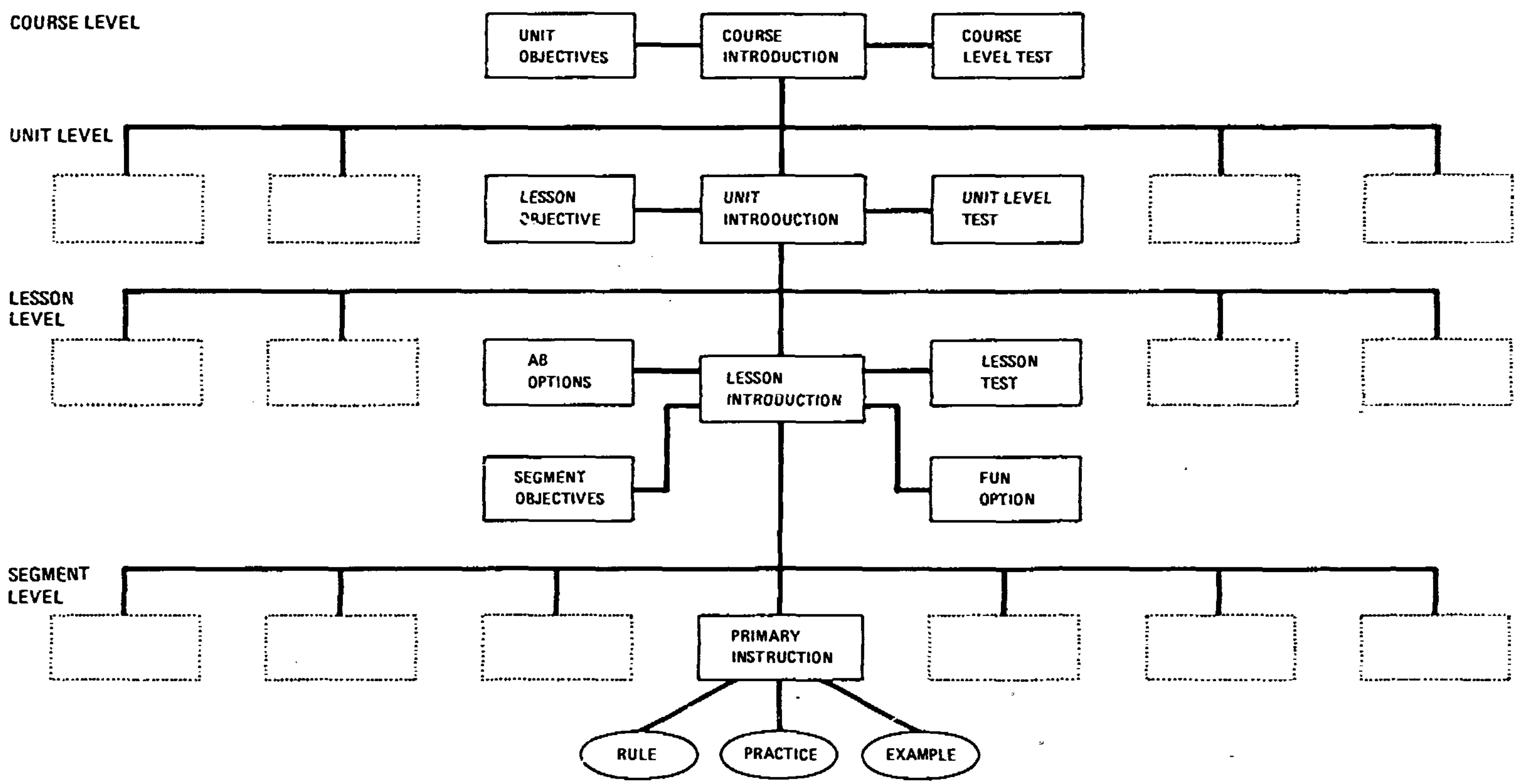


FIGURE 9
COURSEWARE STRUCTURE

Each segment consists of a single generality or rule explained in several styles, a file of instances or examples of the rule, and the number of inquisitory examples or questions that the student may access to personally test his understanding of the concept.

The Learner Control Command Language

The learner control command language allows the student to move through the course hierarchy and access instructional material at his own pace. He may be forced to listen and look at advice if he is going astray, but he is never forced to look at any instructional material that he does not select.

OBJ	HELP	ADVICE
MAP	HARDER	EASIER
RULE	EXAMP	PRACT

At the left is represented the nine principal buttons involved in a learner's control of his own learning tactics. Seven of these deal with events within a segment, while the MAP and ADVICE buttons are more general. The MAP button accesses the next higher level for status or survey, and the ADVICE button elicits advisor program comments on strategy and other matters.

The other seven learner-control buttons above are related to the taxonomy of instructional variables as follows:

OBJ	– Aecesses an illustration of the segment objective.
RULE (Expository Generality)	– Accesses the main generality for a segment.
EXAMP. (Expository Instance)	– Accesses a file of expository instances (with matching, pairing, etc., built in).
PRACT (Inquisitory Instance)	– Accesses the same instance file, but presents them in inquisitory mode.

RULE followed by:

EASIER	– More concrete form of rule (an analogy). Simpler terminology.
HARDER	– More abstract. Technical notation and terminology.
HELP	– Attribute isolation. Definition of sub-concepts. Mnemonic aids; followed by an information processing sequence for using the rule or testing instances of the concept.

PRACT or EXAMP followed by

HELP	– Instance specific attribute isolation to help recall the rule, followed by a step-by-step presentation of a good information processing algorithm for using the rule or testing the concept for this particular example; audio, color and graphics are used.
EASIER or HARDER	– Shifts to easier or harder instances if such are available.

(Note that EASIER and HARDER are "interdisplay relationship" variables while HELP gives "Mathemagenic information".)

Matching of examples and non-examples, and a default sequence generally going from EASY to HARD and covering the necessary range of divergency among the instances is built into the instance files and their controlling logic.

Figure 10a shows a typical lesson level objective and status display (MAP) that a student might access by depressing the MAP key on the learner control keyboard. The numbered boxes represent segment objectives, with their topics listed at the right side of the screen. When more topics or boxes are required, as in this MAP, a second page may be accessed by depressing the space bar. At course level, the boxes on a MAP represent unit objectives; at the unit level, they represent lesson objectives.

The student may survey the (course, unit, lesson) by typing "intro", which accesses a video tape or a minilesson (a filmstrip-like overview). He may also survey the objectives of each numbered box by typing the number of the desired box which, as shown in Figure 10b, highlights the desired box (#2) and then by depressing the OBJ button. Figure 10c is the illustration the student would see for objective 2. If the student types "P" he is provided with a review of the (lesson, unit, course) prerequisites. Status on a MAP is displayed by coloring red, yellow, or green, each numbered box that the student has previously accessed to represent that the student has performed poorly, uncertainly, or acceptably on the material associated with that objective.

On a course or unit level MAP after the student selects a unit or lesson by typing its number, he pushes the GO button (another button on the learner control keyboard) and drops to the next lower MAP. On a lesson level MAP, after the student selects a segment he pushes the RULE, EXAMP or PRACT button to interact directly with the content files.

FIGURE 10a
LESSON LEVEL OBJECTIVE AND STATUS (MAP) DISPLAY

FIGURE 10b
SELECTING A SEGMENT (#2)

The combination of photographing the actual Digicolor display and offset printing requirements result in a less than authentic reproduction of the crisp, high contrast Digicolor display in the screen photographs shown.

FIGURE 10c
REVIEWING THE OBJECTIVE OF SEGMENT #2

Figure 11a shows the rule a student would see following depression of the RULE key on the learner control keyboard. Depressing the EXAMP key the student would see the example shown in Figure 11b. If the student desires an explanation of the example, he could push the HELP key and see the display shown in Figure 11c (which is the first in a sequence of displays the student can move through by pushing ENTER). The student may test his knowledge of the rule with a practice test question by pushing PRACT (see Figure 11d). The bright student can obviously move through each segment spending only a short time while the student having difficulty may look at many examples and take many practice test questions before moving on.

The advisor program provides the student with his status on the course, suggestions on what to do next, and help on how to enter an answer to a question. It limits the student's ability to jump from lesson to lesson indiscriminately. The advice provided is almost exclusively related to lesson tactics rather than content. In addition, the advisor makes no discrimination between students and only bases its advice on the most immediate experience of the student.

At any time the student may request advice by pushing the ADVICE key on the learner control keyboard. Upon doing this he is shown his status (in the course, unit, lesson, or segment level, depending upon where he asks for advice), and a suggestion on what to do next. Figure 12a shows a typical status display at the segment level. The display shows the student how many example and practice items he has worked, and the suggested number he should attempt. To receive a suggestion on what to do next, the student pushes ENTER and sees a display similar to the one in Figure 12b.

The advisor will also interrupt the student and offer unsolicited advice under certain circumstances. Unsolicited advice does not provide a status display but only a message display, as shown in Figure 12c. Advice is given to the student if he is doing something that may be construed as being less productive than a generally effective sequence modeled by the advisor (i.e., if he has failed several practice questions, not reviewed the rule and then asks for a hard practice question). The student may avoid the advice by pushing the key that brought up the unsolicited advice a second time. However, if the student ignores the advice too many times he may see the display shown in Figure 12d.

FIGURE 11a
A RULE DISPLAY

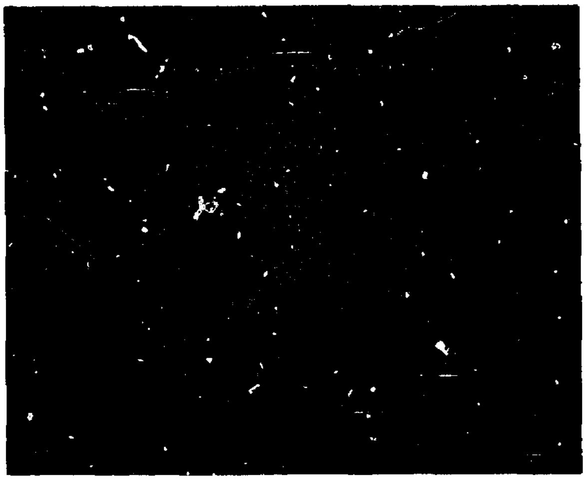


FIGURE 11b
AN EXAMPLE OF THE RULE

FIGURE 11c
THE FIRST PAGE OF A HELP SEQUENCE

FIGURE 11d
A PRACTICE TEST QUESTION

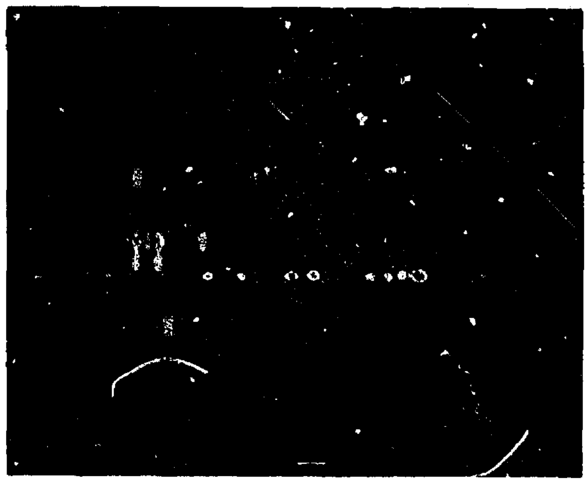


FIGURE 12a
ADVICE STATUS DISPLAY

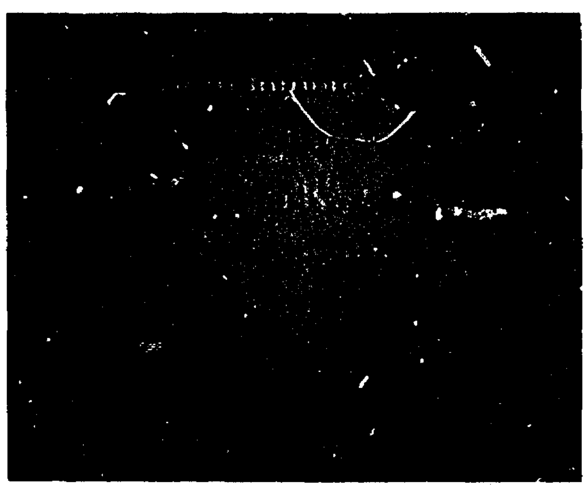


FIGURE 12b
ADVISOR SUGGESTION ON WHAT TO DO NEXT

FIGURE 12c
TYPICAL COMPELLED ADVICE

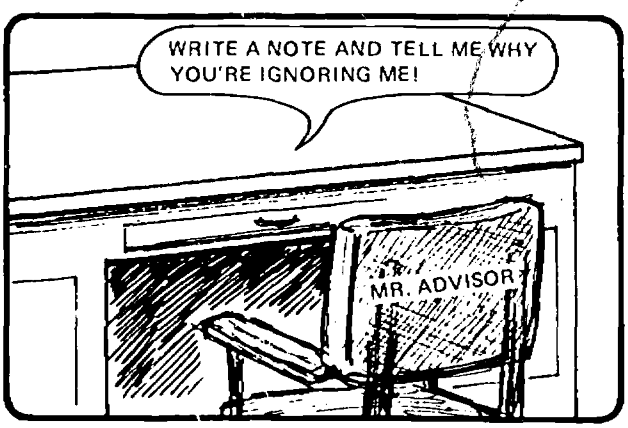


FIGURE 12d
A DISPLAY FOR THE STUDENT WHO IGNORES THE ADVISOR

SECTION V
THE AUTHORING APPROACH

The TICCIT courseware structure separates instructional logic from instructional content. This separation makes possible two innovations in CAI authoring. The first is an especially efficient division of labor to author and prepare instructional content for the system. The second is an authoring system that provides a simplified and convenient means of entering the mass of instructional content but retains much of the flexibility of a general purpose programming language.

TICCIT Authoring Teams

An early goal of the TICCIT project was to develop an approach to CAI authoring that would provide a uniformly high quality product in a timely and efficient manner. The modularity of instructional components offered by the learner control courseware approach opened the way for efficient use of authoring teams to meet this goal.

A separate authoring team has been established at BYU for each of the four courses. Each team is led by the senior author (subject matter expert) and, in addition, includes several assistant authors, an instructional design technician, and an instructional psychologist on a part-time basis. The instructional design technician provides the author with help and ideas on novel ways of presenting material, processing responses to questions, etc. The instructional design technician, typically, has had programming experience and instructional psychology training. The instructional psychologist's role, primarily, is to help the author adapt his material to the learner control courseware structure. For each lesson, teams must produce the following products:

1. Lesson objectives and status display and its contents.
   a. Visualized objectives
   b. Prerequisites
   c. Videotape outline or minilesson
2. Lesson Test and A-B Test
   a. Item formats
   b. Items and sequence

3. Generalities

   a. Rule or definition

   b. Easier version

   c. Harder version

   d. Help for the generality

4. Instance Files

   a. Examples and non-examples, classified as easy, medium and hard, and sequenced appropriately

   b. Instance file formats:

      inquisitory format (practice)
      expository format (example)
      help format
      feedback message format

   c. Answer-processing specifications for practice instances

As shown in Figure 13, two other groups work with the authoring teams in the production of CAI lessons. The quality control committee reviews each lesson prior to packaging, and the packaging group prepares each lesson for input into the computer. The quality control committee consists of a subject matter expert, an instructional psychologist, a graphic specialist, and an editor. The lessons are assigned for review to members of the committee who were not involved in the lessons' original production. The function of the reviewers is to make suggestions and recommendations to the senior author with regard to the subject matter included in the lesson, the proper use of instructional variables, the lesson format, the grammar and composition used in the lesson, and the suitability of the suggested graphics.

The authored drafts are converted into a form ready for computer processing in the packaging group. The packaging group double checks the material the team has authored to catch mistakes in specifying locations of displays, types of answer processing, etc. In addition, camera-ready graphics are produced and the authored lesson and graphics are entered into the computer.

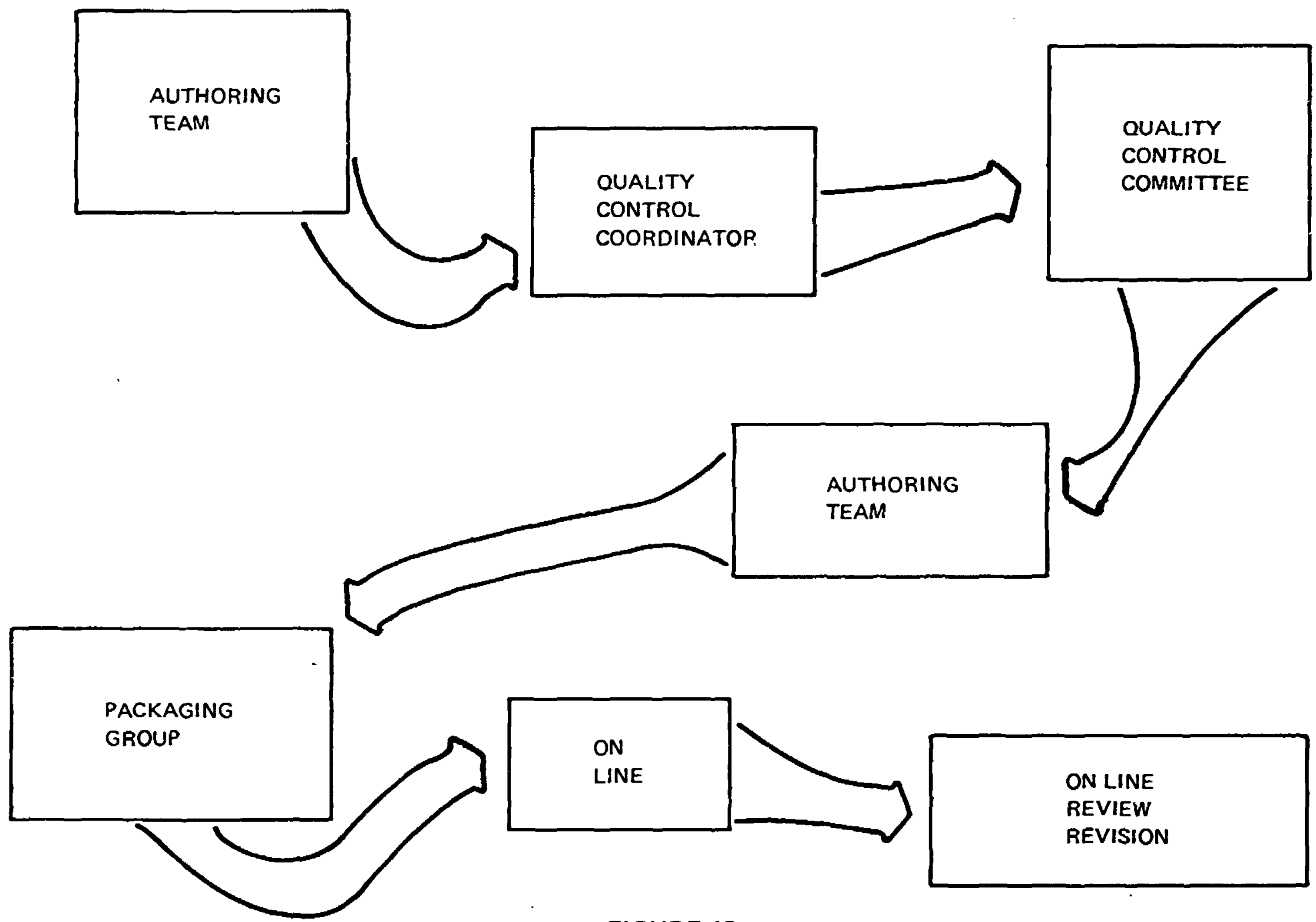


FIGURE 13
TICCIT PRODUCTION FLOW

The lesson production schedule, basically, is to produce all of the components and their display formats for a complete lesson (except videotapes and AB options) in a five to seven-week cycle. The main content files for a lesson are produced in two or three weeks. The next two weeks are dedicated, primarily, for review and revision; and, the final one to two weeks are provided for packaging. On each authoring team a new lesson enters this five-week's cycle every week.

The Authoring System

The TICCIT Authoring System utilizes forms which are filled out by the authors and the packaging group to indicate the types of displays that are presented to the student. The TICCIT system allows authoring to be done either on-line or off-line. In the off-line mode the authoring team may write the data on paper forms for later entry into the system. After the forms are filled out the data entry operation takes place. An operator sits at a TICCIT terminal and enters the data into the system. The data entry system sequentially places images of the forms on the screen. The forms in the computer and the paper forms are so correlated that the operator need not look at the screen in order to enter the data. It is expected that the operator will look at the screen when a page has been typed to visually verify the data. The data is entered into source files in the computer.

In the on-line authoring mode the identical programs can be used to generate courseware material. Instead of an entry clerk sitting at the terminal, an author may use the computer-generated forms instead of the paper forms. The author then proceeds to fill them out in the same manner as he would using paper forms. All of the data is then entered into the source files. When other members of the authoring team wish to add examples and practice problems to a segment, they first call up and look at the displays that have been specified. They then add examples and practice problems to the source files. This approach permits one authoring system to be used both in an on-line and an off-line mode.

To illustrate how the authoring system operates, let us take a look at an on-line author entering a segment into the system. The author selects from the system menu the data entry system. He specifies the lesson and segment that he wishes to work on. The first display of a segment that is presented to him is the rule form (Figure 14a). The rule form is, in essence, a blank sheet of paper on which the author constructs the rule display. The author using his TICCIT keyboard constructs the display on the screen. He types the text in the proper place in the proper color (Figure 14b) and pushes the ENTER key. There are similar forms that are used for entry of an easier statement of the rule, a harder statement of the rule, and a help for the rule.

Next comes the entry of the format of the example and practice displays. Figure 14c shows the display screen containing the display specification form. In the upper right

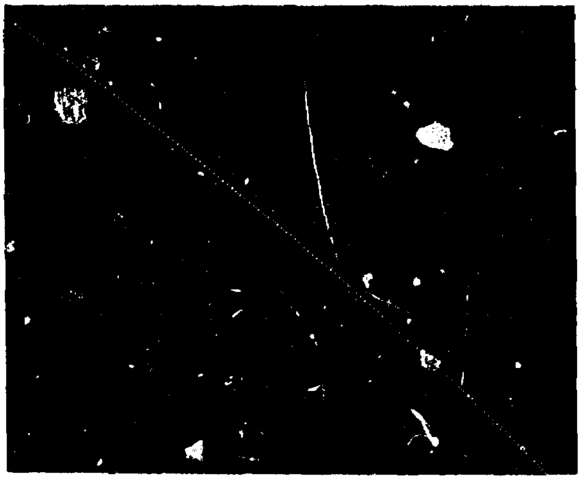


FIGURE 14a
RULE FORM

FIGURE 14b
RULE FORM FILLED IN

FIGURE 14c
DISPLAY SPECIFICATION FORM

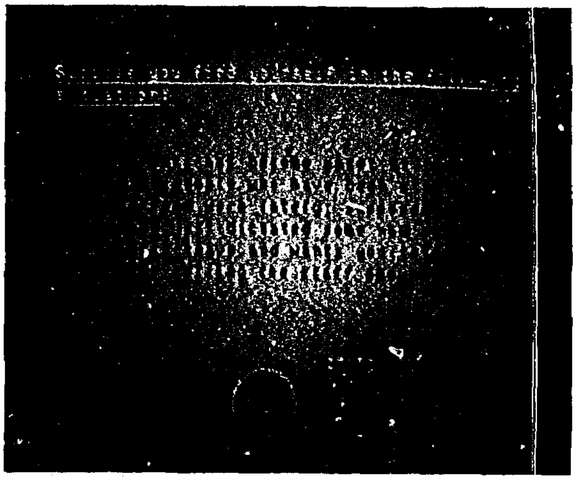


FIGURE 14d
EXAMPLE OF BASE FRAME DATA

of the form the author specifies whether there is one page or multiple pages to this example. The author specifies whether or not he wants the complete screen or part of the screen to be erased before the display is shown to the student. If the author wishes to put a timed display on the screen, he puts the number of seconds he wishes the display to remain on the screen (for a student) in the appropriate box. Next the author will specify the format of the display. This is done by dividing the screen into author-defined windows. The screen size is 17 rows by 43 columns. The individual rows are lettered A through S, and the columns are numbered from 1 to 43. The author gives each window a label that is an alphabetic character. Then he specifies the upper left and the lower right coordinates of the display window. All the display windows are rectangular.

The author next specifies where information is to come from to fill in the windows. There are three places where information can be obtained for filling in windows. The first is from the Base Frame file. The Base Frame file contains text that will be used on many different displays within the segment. For example, the message underlined in Figure 14d may appear in all example displays in this segment. It is placed in the Base Frame file and stored only once in the system. Other text that is used only once in the segment is stored in the Instance file. The author indicates which file he wishes to use by placing a B or I in the file column. Information may also be obtained from the Graphics file. It may be either from Instance Graphics (IG) or from a Base Frame Graphics (BG) file. All their files are divided into items and states. An item corresponds to one example or to a practice problem. States within a file are different elements of information within an example or practice display frame. In the Instance file there is a variable number of items. The system, when it is execution mode, will successively go through item after item as requested by the student.

The justification code determines how the textual information is to be spaced in the window. It may be top justified, left justified, right justified or bottom justified.

In filling in this display the author has determined the format of all example displays for a particular segment. A similar form is used for the help displays, and the practice displays within a segment.

When it comes time to fill in the data to be placed for the example display, a form as in Figure 14e is shown to the author. This form delimits the window that he has specified. The author can enter in seven colors the information he wants to place on the screen. The use of this form guarantees that the information will fit into the specified window. As stated before, an author may fill in the information for one or two examples and practice items and let other project personnel fill in the remainder of the 10 – 30 items.

FIGURE 14e
INSTANCE FILE FORM

FIGURE 14f
ANSWER PROCESSING FORM

A special form is used to specify the types of answer processing that are to take place for the practice problems. An example of this form is shown in Figure 14f. The types of answer processing that can be specified by using this form are: string matching (strings of words), either ordered or unordered, and with misspellings allowed or not allowed. Matching of algebraic expressions can be handled, as well as student inputs that are in several types of set theory notation.

SECTION VI
SOFTWARE

The TICCIT software system is divided into three parts: the operating systems, the application software, and the authoring software. The operating system is a special purpose operating system designed specifically for the delivery of CAI to more than 100 terminals simultaneously. A general purpose operating system would not be efficient enough for the task. The operating system software is concerned with task scheduling, disc management and message handling. The applications software runs under the operating system and provides the specific CAI functions. The authoring software is, in reality, a specific collection of special applications programs. Any terminal can be used for on-line authoring or for courseware execution.

TICCIT Operating Systems

The primary purpose of the TICCIT system is to present Computer-Assisted Instruction (CAI) efficiently to over one hundred students simultaneously. Furthermore, each of the students using the system is to be independent of the other students with no necessity for any student to be at the same place in any course as any other student.

The major problem in the system is how to run the courseware programs and the data for the more than one hundred students in a minicomputer and maintain a response time of typically .5 seconds. Analysis of the TICCIT operating systems shows that more than 100 simultaneously active terminals can be supported. When the system is fully loaded, 79 milliseconds of processing time are available for each student input. In 79 milliseconds 49,000 instructions can be executed. Experience of other CAI systems indicates that this provides a comfortable margin of safety.

The major functions of the operating systems are:

- Echoing of the keyboard strokes on the display.
- Editing of the input messages.
- Providing student timing information.
- Coordination of the delivery of audio, text, and graphic information to the student.
- Accessing programs that perform answer processing of the student input in order to select the proper courseware program (which, in turn, do housekeeping and generate new displays).

- Swapping courseware programs and user data in and out of memory.
- Scheduling the execution of the courseware programs.
- Providing storage for student data.
- Maintaining student registration data.
- Providing a procedure for orderly startup, startover, and shutdown.

The approach to the problem has been to divide the functions of the operating system between two processors as shown previously in Figure 3. The terminal processor handles all of the input and output between the student and the system. The main processor only sees complete messages that are ready for processing and does not have to devote overhead time to editing messages or to the housekeeping tasks that are associated with making efficient use of a variety of delivery equipments. This basic division of functions plus the organization of the user programs and student data in the main processor enable two minicomputers to support such a large number of simultaneous users.

A user authorization file is maintained on the fixed head disc. This file contains a record of which students are registered for each course and also contains the student's password. Another disc in the system contains student permanent data records. There is one student data record for each course in which a student is enrolled. The student data record contains information that shows the progress of the student through each course and exactly where he is in the course at any given time. When a student logs on the system, a copy of his data record is placed on a disc which is organized to facilitate fast swapping in and out of memory.

As a student progresses through a session, the student data record records his progress. Periodically, the temporary copy of the data record is placed in the permanent file (a checkpoint). This feature protects against all of the work that the student has done being lost in case the system goes down because of power failure or certain hardware or software problems.

When the system goes down for any reason, and the discs are still intact, all students resume their session from the record contained in the permanent student record which reflects the last checkpoint. Checkpoints are taken at logical or educationally meaningful points.

Protection against software errors in the courseware programs is provided by the memory protection features of the minicomputer. Before a courseware program is executed, a core memory mapping function takes place so that all courseware programs appear to operate in the same location. In addition, read and write protection is invoked so that a particular program cannot read or write core outside of its assigned area. This means that a courseware program error has limited effects. A program error affects only one terminal user and cannot interfere with the operating system.

When a courseware program begins execution, a timer is started. A "time-slice" is used to prevent one student from using more than his share of the computer time. When a time-slice runs out, this student's job is put at the end of the input queue. The time-slice duration is a system parameter which can be adjusted to tune the system. Experience with CAI systems indicates that a time-sliced termination of a courseware module will be infrequent.

All courseware programs are divided into modules of 4000 words (8K bytes) or less. Each module is re-entrant and contains pure code and constants. All data that is student-specific or is subject to change is kept in a student data record. This division means that while both programs and data must be brought into core for program execution, only data must be written out of core and saved between program executions. Referring to Figure 15, it can be seen that the core of the main processor is so divided that there are five user areas. The operating system allocates time between the areas so three student data areas and two courseware program areas are in core at one time. The answer processing programs use one student data area to process a student's response and thus determine the next program to be loaded. At the same time a program is executing, using a program area and a student data area, the third student data area is being swapped out to disc and a program is loading in the second program area. The timing chart for this cycle is shown in Figure 16.

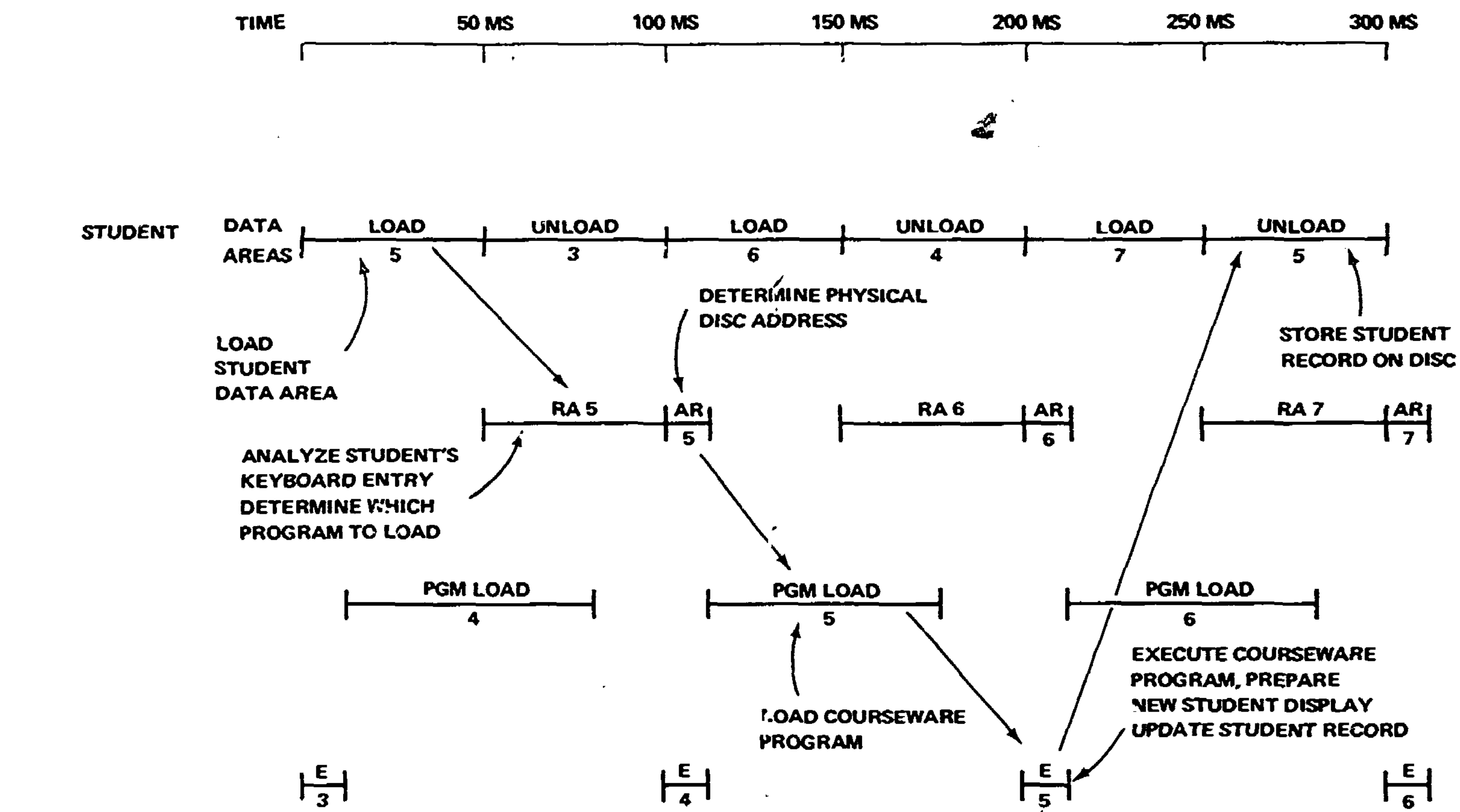
To make the simultaneous loading of programs and data possible, two disc channels are used so that the load operations do not interfere with one another.

The organization of the control programs in the main processor is illustrated in Figure 17. The main control program is the heart of the operating system. It passes control to the scheduler which accepts student input messages and forms a task queue. The main control program next uses the loader to bring the programs and data areas into core for the next task in line. While the loading operation is going on, the execution of the previously loaded module is in the other partition. When execution is complete, the data area is swapped out.

The movement of program modules into processor storage and of student data records in and out of processor storage is the major job of the main processor operating system. This operating system also supports input/output operations by the courseware modules (e.g., output to the log tape, input/output to the student terminals via the terminal processor) and handles errors in the courseware programs.

STUDENT DATA AREA 8 K BYTES
COURSEWARE PROGRAM AREA 8 K BYTES
STUDENT DATA AREA 8 K BYTES
COURSEWARE PROGRAM AREA 8 K BYTES
STUDENT DATA AREA 8 K BYTES
ALGOL LIBRARY 12 K BYTES
APPLICATIONS PROGRAMS 20 K BYTES
OPERATING SYSTEM 24 K BYTES

FIGURE 15
MAIN PROCESSOR CORE ALLOCATION



NOTE: THIS FIGURE SHOWS OVERLAP OF MAIN PROCESSOR OPERATIONS ON 5 ARBITRARY STUDENTS (#3, #4, #5, #6, AND #7)
THE COMPLETE EXECUTION CYCLE OF STUDENT #5 IS HIGHLIGHTED

RA	RESPONSE ANALYSIS
AR	ADDRESS RESOLUTION
PGM	COURSEWARE PROGRAM
E	COURSEWARE EXECUTION

FIGURE 16
MAIN PROCESSOR EXECUTION CYCLE

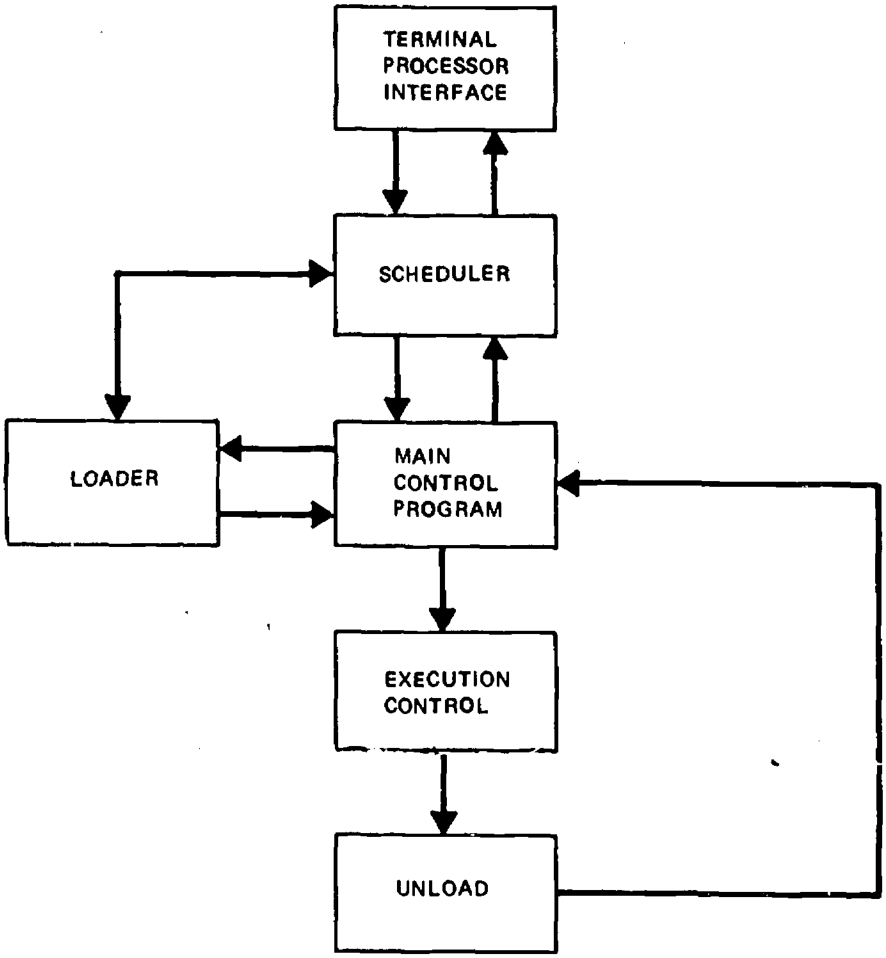


FIGURE 17
MAIN PROCESSOR PROGRAM ORGANIZATION

An instrumentation package has been designed into the main processor operating system from the very beginning (see Figure 18) because a capability to record all inputs and outputs on magnetic tape can provide a valuable debugging tool to test the proper functioning of the operating system and the courseware. The operating system itself uses an instrumentation program, which can be made to operate between all other programs. This allows for system traces, system timing, and selective looks at critical system parameters. In addition, the courseware programs themselves do internal data recording and place information on magnetic tape.

The terminal processor handles the detailed hardware interactions. As the student types an input (1 to N characters), the terminal processor displays the characters on the student's TV (this is called "key echoing"). The terminal processor also examines each character to see if it is an end-of-message. When an end-of-message is received, the complete message is code converted as necessary, edited, and then passed as a complete unit to the main processor for courseware processing. After processing, the main processor sends the reply to the terminal processor. Text messages are formatted and the proper hardware control information is added. However, close control of output is required so that textual, graphics, and audio output reach each student in exactly the sequence desired by the authors of the courseware. The terminal processor supplies both pre-processing and post-processing functions for all messages that enter and leave the TICCIT computer.

Application Software

The Application Software consists of the computer programs that implement the courseware. These programs analyze the student input, keep track of where the students are in the courses, construct the display to put on the student terminal, and perform other computer related functions. The Application Software is divided into the following areas:

- Answer Processing
- Advisor
- Logging
- Primary Instruction
- MAP
- Test Logic

The applications programs are used to process all student inputs. For processing efficiency they are all written in assembly language and are all core resident.

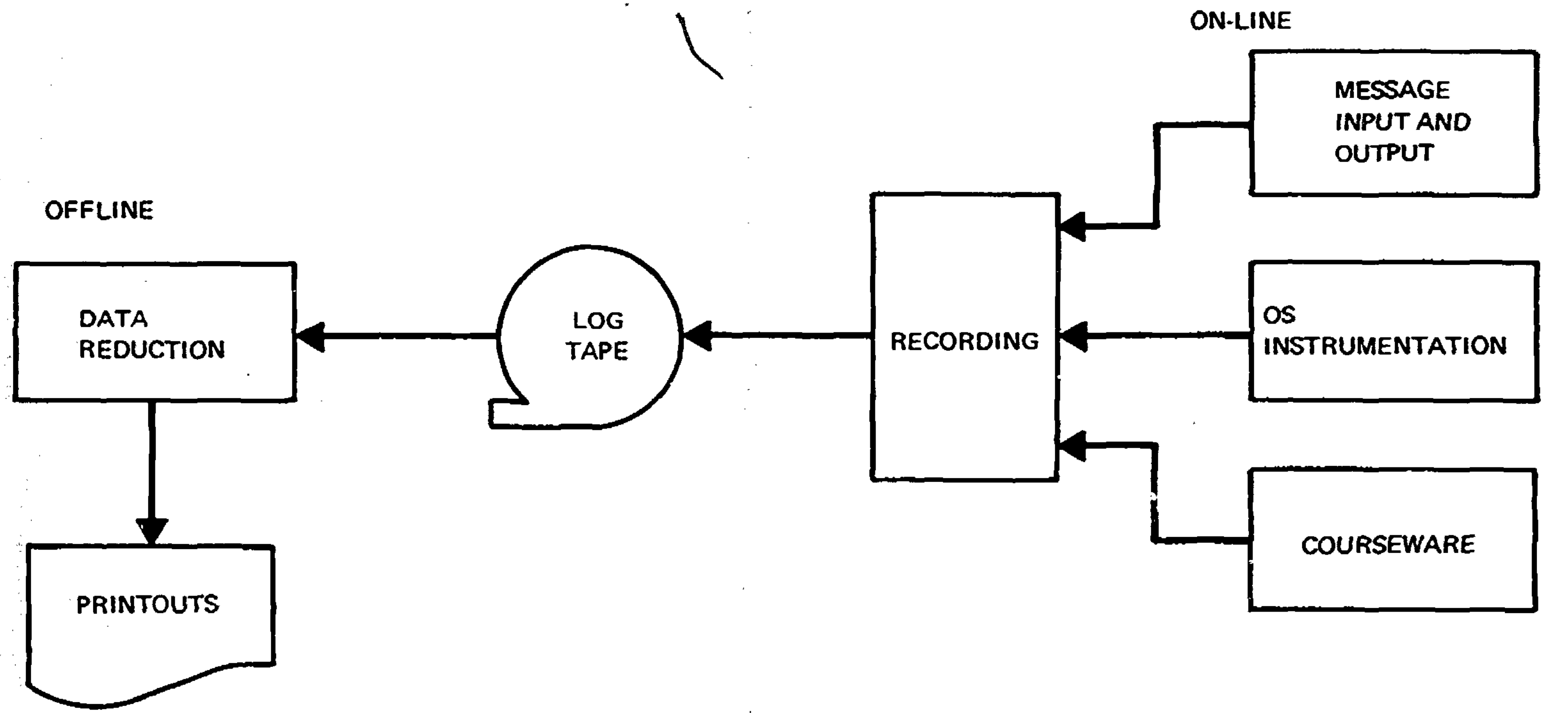


FIGURE 18
MAIN PROCESSOR INSTRUMENTATION

The answer processing routines provide special functions needed to analyze student responses. The lexical routines divide a response into words and punctuation and recognize special lexical types. Examples of special lexical types are; yes-no responses and letters when multiple choice response is expected. The string matching routines match a student's input with a specified response. In addition, they determine whether or not the word in a response is in the proper order or out of order and unless deactivated by the author, they automatically check for misspelling. The string matching routines also set special system variables so that different feedback displays can be given to the student if part of the answer is correct, but the entire answer is not present. The "set builder" routines recognize students inputs that are in set builder notation (e.g., the set of all X such that X is greater than 1 and less than 6.) The roster routines recognize a student's input that is in set roster notation. This means that all elements of a set are listed. The expression evaluation routines operate on numerical expressions and functions such as log, natural log, sine, cosine, tan, arc tan, and absolute value. Which of the above routines operate on a given student input is decided by the author's parameters on the answer processing forms. An example of the answer processing forms that has been filled out was shown in Figure 14f.

During processing of the source data prior to running students the information on the answer processing form is analyzed and encoded in a compact form designed for efficient manipulation. In real time the student's input must be analyzed and encoded in an identical manner. Internally the words and their lexical types are represented by a set of pointers. The program gains efficiency by primarily manipulating the pointers and not the words themselves.

The Advisor program is a state-driven program. The structure of a course can be divided up into a series of states depending upon whether the student is currently operating within a lesson, at the lesson overview level (looking at the Lesson MAP display), the unit level or the course level. There are well defined sets of rules for moving from one part of a course to another and thus for moving from one state to another. When the student is in a given state the advisor program has only a few tests to make to determine whether or not unsolicited advice must be given to a student. Subdivision of the program logic into many states, each of which have only a few tests, contributes to system efficiency.

The logging program puts information on magnetic tape for non-real time data reduction. This program operates for every student input and puts the following types of information on tape.

- Time

- Student Latency (time between last output to the student and his input)

- Results of Answer Processing including error indicators

- Student Input

Options in the program permit the system operator to turn off logging or to log less data or to log only certain terminals.

The primary instruction programs keep track of where the student is within a lesson by setting indicators in the Student Data Area. A very detailed record is kept of the student's progress in the current and one previous lesson. Summary data is kept for all other lessons. For example, a record is kept of how many difficult practice problems the student has answered correctly in segment X of the current lesson. The primary instruction programs also assemble parameters that define the next display, sending it to the student via an output program and the terminal processor.

The MAP program maintains variables that determine the student's progress from lesson to lesson and unit to unit. These variables are also kept in the student data area. Some of the variables determine the color of the boxes on the MAP display. A box is set to green color if the student has done well on a given segment of a lesson, yellow if so so, and red if poorly.

The Test programs put the student in a special mode and inhibit the operation of certain other system functions. For example, when a test question is answered a feedback display is not given immediately to the student. Using parameters set by the author and included in the test load module a determination is made as to when a student has passed or failed a test. The test program records the test scores in the student data area.

TICCIT System Analysis

The parameters that are most important to system responsiveness and system throughput have been studied both by design analysis and computer simulation. Both methods indicate that expected levels of performance are reasonable and will be attained.

The design analysis approach uses the current system design and calculates the processing time available for courseware.

The basic program execution cycle in the TICCIT system is 100 milliseconds (ms). This cycle is arrived at by using the following assumptions:

- There are 128 active terminals.

- There is one input per terminal every 13 seconds.

- Each input is 10 characters long.

- An average of 20 characters per input is placed on the system logging tape.

- Each output to a student averages 100 characters. 100 students logon and logoff the system per hour.

Based upon information gathered from other CAI systems, these assumptions are conservative. In many cases the average time for a student to respond will be 20 to 25 seconds. Using the current design of the TICCIT delivery system and, in particular, the design of the operating system of the main processor, an analysis of throughput of the system has been done. In the 100 ms program cycle there are 125,000 machine cycles. Of this total 13,000 machine cycles are used for input and output. This includes all information transferred to disc, to tape, and to the other processor.

Input/output represents 10 percent of the machine cycles. The operating system executes 3700 instructions per 100 ms cycle. Since each instruction requires an average of two machine cycles, the operating system uses 6400 machine cycles or 5 percent of the machine cycles. The addition of the input-output and operating system figures gives the total system overhead of 15 percent. This leaves 85 percent of the time available for courseware execution. In this time, 49,000 instructions can be executed. The experience of other CAI systems indicates that this is more than enough time to do traditional CAI functions. In fact, nontraditional functions of TICCIT, such as computer aided grading of English compositions and providing individualized advice to the student for learning strategies, can be accommodated. The TICCIT design analysis indicates that significant processing per input can be accomplished for each student.

The computer simulation was done by creating a model of both processors in the TICCIT delivery system. This model simulates the actions in each processor and the many interactions involved in handling each student. The data from each student are represented as the characters received when the student types on his keyboard, as the resultant message for processing by the courseware and as the new display produced by the courseware.

The major measures of interest are:

- The delay between the time a student types a key on his keyboard and the time the character is displayed on his terminal screen (the key echo time), and

- The delay between the time a student completes typing a message and the time he receives a response from the courseware.

The model, as shown in Figure 19, estimated the key echo time at less than 1/4 second, and the courseware response time at about 1/2 second with a full load of students. This rapid responsiveness should meet the student's needs with a considerable margin of safety.

The model itself is a sizable computer program. IBM's General Purpose System Simulator (GPSS) was used to develop the model. Each character typed by each student is handled separately in the terminal processor section of the model. The rate at which students type is varied with some randomness to reflect the differences experienced with actual students at the University of Texas CAI Laboratory. The number of characters in each message also has random variation representation of the anticipated characteristics of the courseware.

Throughout the model, the student's data passes through the same steps (and delays) that it will encounter in the actual system. The accumulation of times in these steps and delays for many students' messages provides the statistical base for estimating the performance of the system.

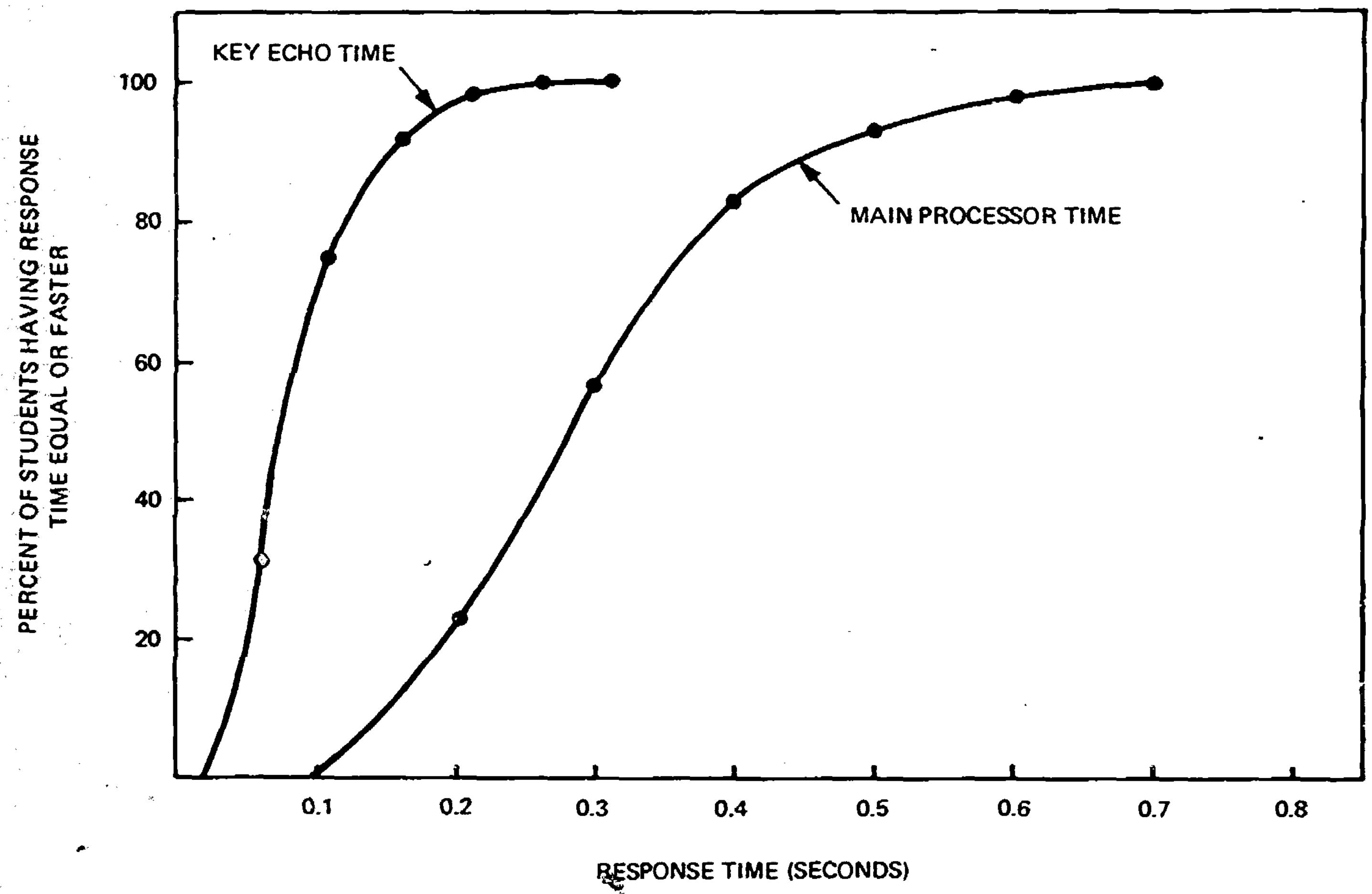


FIGURE 19
SIMULATION MODEL RESULTS
128 STUDENT LOAD

SECTION VII
IMPLEMENTATION OF TICCIT IN COMMUNITY COLLEGES

An analysis of the reasons for failure in previous efforts to market CAI have invariably led to the identification of hardware, software, and courseware components as being crucial. Hardware has not provided the rich display capabilities required by education as contrasted to business and science. It has not been able to provide these display and processing capabilities at a low enough cost. Software has not been amenable to the production, revision and testing of educational materials. Finally, a body of courseware sufficient to make a major impact in an educational curriculum has not been available on cost-competitive systems. Solutions to all three of these problems, however, can still fall far short of meeting the goals of this project. At least as important as the hardware, software, and courseware, is the implementation plan and procedures which are followed to introduce a new system into a venerable educational establishment, with all of its traditions, opinions, philosophies, personalities, and power groups. It has been quite apparent with attempts to introduce other forms of technology, and especially individualized instruction, that they do not fit well. An instructional system engineered for self-paced progress, effectiveness, efficiency and a redefinition of teachers' roles is a foreign body being introduced into an educational system. As in the case of an organ transplant, there can very often be a rejection which will destroy the transplanted subsystem.

In the proposal initiating this project, this problem was discussed with emphasis on only one aspect of the implementation; that is, the selection of two appropriate sites. A set of criteria was listed which included a readiness for innovation within the college administration and faculty, the system in which the campus was embedded. For example, there are states which have regulations which greatly restrict the use of computer equipment. There are states which pay for instructional costs by means of a classroom contact hour base. This may work as a disincentive to getting students through more efficiently. In these states, students cannot be awarded the same credit if they spend less than the indicated amount of time in class. In the site selection criterion, there was nevertheless an attempt to find schools which were representative and which had a fairly diverse student population, so that the limits of the system could be fairly evaluated. Representativeness was seen, however, as being less important than the leadership capability and the positions of the schools within the junior college movement. Another criterion was a pragmatic one. We sought to select two campuses which were geographically proximal, one to Brigham Young University, the other to MITRE Corporation.

A number of campuses were identified for further investigation and the choice was finally narrowed to the Maricopa County Community College District in Phoenix, Arizona and the Northern Virginia Community College. The site selection was accomplished by

The MITRE Corporation in close cooperation with Brigham Young University and Educational Testing Service. After high-level meetings with the administrators of both systems, the Phoenix campus of the Maricopa district and the Alexandria campus of the Northern Virginia district were selected.

A letter of agreement was signed by representatives of each community college and negotiation of a contract laying out the details of the relationship between The MITRE Corporation, BYU and Educational Testing Service was drafted and consummated. The contracts specify the conditions under which the community colleges would cooperate in the installation, demonstration and evaluation of the TICCIT systems during 1974-1976. This demonstration required the commitment of funds from the community colleges for physical plant modifications and staffing to accomplish the demonstration and evaluation. It also provided that they would make budgetary and other information available to Educational Testing Service for a very thorough evaluation of costs and effectiveness at several levels of consideration. Because of this commitment of funds, and because of the planning and leadership which have been forthcoming from the community colleges, these two colleges are, in a very real sense, partners in this entire project.

Implementation Plan

During the contract negotiations with the community colleges, heavy stress was laid upon the development of a comprehensive and valid implementation plan. After considerable communication and cooperative effort between MITRE, BYU, ETS, and the two community colleges, the following planning areas were specified. Each major planning area is covered in a separate section of an implementation planning document.

IMPLEMENTATION PLANNING DOCUMENT

Section A. Management and Organization

Responsibility: MITRE

This section contains material related to the different organizational structures of the agencies involved in the project. It also specifies management procedures and coordination policies.

Section B. Schedule of Activities

Responsibility: MITRE

Major project milestones are recorded in this section. Individual schedules for such things as courseware development, content validation, training and orientation, equipment installation, etc., can be found here.

Section C. Support Requirements

Responsibility: The Community Colleges

This section is intended to assist the community colleges in cost compilation. All factors which have an impact on college resources can be included in this section.

Section D. Facility Preparation

Responsibility: The Community Colleges

Such things as floor plans, air conditioning plans, light and power diagrams, and carrel specifications can be found in this section.

Section E. Training and Orientation

Responsibility: BYU and Community Colleges

Section E will describe the training and orientation of community college administrators, teachers, TICCIT proctors and students. It describes the events which will occur during the production and demonstration phases of the project.

Section F. Courseware Development Coordination

Responsibility: BYU

This section contains a description of how the community colleges and BYU will assist each other in the development of TICCIT courseware.

Section G. Curriculum Integration

Responsibility: The Community Colleges

A description of how TICCIT courseware will be used at each community college is found in this section.

Section H. Installation and Check-Out

Responsibility: MITRE

A detailed plan related to the movement and installation of TICCIT hardware is the subject of this section.

Section I. Operation

Responsibility: The Community Colleges

This section covers several matters related to the actual operation of the system. Descriptions of teachers' roles, proctors' roles, etc., can be found in this section. Student procedures, course scheduling and system operating instructions are also discussed in this section.

Section J. Contingencies

Responsibility: The Community Colleges

Plans related to various contingencies are contained in this section. Matters such as who will make decisions, what actions will be taken, and what reports and follow-up actions are required are also specified in this section.

Section K. System Maintenance

Responsibility: MITRE

Descriptions of system maintenance routines, policies, and procedures are contained here.

Section L. Formative Evaluation

Responsibility: BYU

This is a detailed explanation of how BYU plans to evaluate its courseware before it is sent to, and while it is at, the community colleges.

Section M. Summative Evaluation

Responsibility: ETS

This is a detailed description of the evaluation plans of Educational Testing Services (ETS).

Section N. Visitor Control

Responsibility: The Community Colleges

A plan which explains how visitors will be briefed and controlled at the community colleges will be recorded in this section.

Section O. Supporting Documentation

Responsibility: N/A

This last section contains a listing of documentation required in support of the implementation and demonstration of the TICCIT project.

The plan was published by MITRE on October 1973. The plan is meant to be flexible so that it can evolve as the project impact on the community colleges becomes better understood.

Community College Authors at BYU

Even before the first draft of the plan was produced, it became apparent that both courseware production and implementation could be greatly enhanced by the presence at

BYU of faculty members from the Mathematics and English Departments of Phoenix College and Northern Virginia College. Steps were taken to accomplish this during the fall of 1972, and in January of 1973, a mathematics faculty member and an English faculty member from each of the two colleges joined the four BYU courseware production teams. These individuals have made a signal contribution to the courseware development. Their insight into the range and ability and background in the students, and into the way instruction is organized at the two community colleges is helping to shape the courseware as it is produced. They are actually writing parts of the courseware as members of the production teams.

Of equal importance to the community college authors' contribution to the courseware development is the contribution of the community college faculty to the development of the implementation plan. Faculty procedures include a definition of new roles for faculty members who will still be involved in the administration of the mathematics and English courseware. While it is expected that there will be fewer faculty involved for the same number of students, it will still be necessary to have faculty in both courses. For example, in the English course a writing laboratory may be provided and faculty members will grade the written productions of students which occur at different points during the English lesson material. After grading the writing samples of the students according to a checklist of categories, the faculty members enter this material into the computer so that a prescription system can recommend the next lesson the student will take. Faculty members and proctors also assist students when they have failed a lesson test two times. The third time the student is taken through with a person. Another part of the implementation plan is related to the amount of credit assigned to each course, the number of students who will be enrolled, and proctor procedures.

Upon their return to Phoenix and Alexandria, the community college authors have served as an important liaison function during implementation planning. Initial experience has shown that a system engineered as the TICCIT courseware is a complete man-machine system. It is difficult and complex to grasp by faculty. In addition, it introduces a great deal of threat because it offers the possibility that administrators will choose to reduce faculty rather than increasing enrollment or differentiating course offerings. Close coordination between all parties throughout the entire project is required to keep the administration and faculty well informed and enthusiastic.

THE MITRE CORPORATION